UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Non-convex Optimization in Machine Learning: Provable Guarantees Using Tensor Methods

Permalink

https://escholarship.org/uc/item/7p90p57n

Author

Janzamin, Majid

Publication Date

2016

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Non-convex Optimization in Machine Learning: Provable Guarantees Using Tensor Methods

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Majid Janzamin

Dissertation Committee:
Professor Animashree Anandkumar, Chair
Professor Athina Markopoulou
Professor Padhraic Smyth

2016

© 2016 Majid Janzamin

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I am truly thankful to my advisor, professor Anima Anandkumar, for her support and guidance over these years. She has set an excellent example as a researcher, mentor and teacher for me, and she has always inspired me over the years of my Ph.D. research.

I would like to thank my Ph.D. committee members, professor Athina Markopoulou and professor Padhraic Smyth who valued my research and gave me invaluable advice for my future career. I would also like to thank professor Max Welling who was on my Ph.D. candidacy committee and admired my research.

I also thank my collaborators, Rong Ge, Daniel Hsu, Sham Kakade and Hanie Sedghi. It has been a fun and productive experience working with them.

I would like to thank University of California, Irvine and Department of Electrical Engineering and Computer Science for providing the infrastructure and research environment to help me complete my dissertation and supporting me by EECS department fellowship.

This dissertation has been supported by NSF awards FG16455 and CCF-1219234, ARO award W911NF-12-1-0404 and ARO YIP Award W911NF-13-1-0084.

Chapter 5 and portions of Chapters 2 and 3 were first published in Journal of Machine Learning Research. I would like to acknowledge Journal of Machine Learning Research for giving me the permission to incorporate my publications in this dissertation.

A special feeling of gratitude to the friends who supported me in every respect through this journey, especially Afshin Abadi, Mohammad Abbasi, Hamed Abrishami, Mohammad Reza Aghajani, Amin Amouhadi, Sanaz Barghi, Hadi Goudarzi, Alireza Imani, Sina Jafarpour, Fatemeh Kashfi, Salman Khaleghi, Sina Khaleghi, Sanam Mirzazad, Ali Moayedi, Mohammad Naghshvar, Parastoo Qarabaqi, Tooraj Rajabioun and Roozbeh Tabrizian.

I wish to acknowledge help and advice from my dear friends and colleagues, Taha Bahadori, Behnam Bahrak, Ahmad Beirami, Adel Javanmard, Mohsen Karimzadeh, Hamed Maleki, Rina Panigrahy, Rodolfo Victoria, and Li Zhang.

Most importantly, I would like to thank my family. My wife, Hanie Sedghi, who has been the source of love, support, encouragement and motivation over these years, spurred me on with her patience and this work would not have been possible without her; and my parents Mehri Ashouri and Asadallah Janzamin, for giving me life, love, support and constantly encouraging me to pursue my education; and my siblings Katayoun and Mohammad, for their encouragement and support. I am also grateful to my grandmother, Shokat for her love; my parents-in-law, Mitra Soraya and Kambiz Sedghi, and my siblings-in-law Hosna, Hoda, Mehdi and Vahid for their support.

CURRICULUM VITAE

Majid Janzamin

EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering — **June 2016**
University of California, Irvine — *Irvine, CA*

Master of Science in Electrical Engineering — **January 2010**
Sharif University of Technology — *Tehran, Iran*

Bachelor of Science in Electrical Engineering — **August 2007**
Sharif University of Technology — *Tehran, Iran*

Work & Academic Experience

Graduate Research Assistant — **Sept. 2010–June 2016**
University of California, Irvine — *Irvine, CA*

Research Intern — **July-Sept. 2014**
Microsoft Research Silicon Valley — Mountain View, CA

Visiting Graduate Student — **Sept.-Oct. 2012, April-June 2014**
Microsoft Research New England — *Cambridge, MA*

Visiting Graduate Student — **Sept.-Nov. 2012**
ICERM, Brown University — *Providence, RI*

Member of Technical Staff — **Nov. 2007–Sept. 2010**
Parman Co. — *Tehran, Iran*

TEACHING EXPERIENCE

Co-instructor of Detection and Estimation Theory — **Winter 2014**
University of California, Irvine — *Irvine, CA*

AREAS OF INTEREST

- Machine Learning, Data Science and Statistics
- Convex and Non-convex Optimization: Analysis and Applications
- Inference and Learning in Graphical Models and Latent Variable Models
- Spectral Methods: Tensor Decomposition Analysis and Applications

PUBLICATIONS

Note: authorship order of specified publications by **asterisk symbol** * follows the convention in theoretical computer science of **alphabetical author order**.

REFEREED JOURNAL PUBLICATIONS

"When are Overcomplete Topic Models Identifiable? Uniqueness of Tensor Tucker Decompositions with Structured Sparsity", by A. Anandkumar, D. Hsu, M. Janzamin* and S. Kakade, *Journal of Machine Learning Research*, 16:26432694, Dec. 2015.

"High-Dimensional Covariance Decomposition into Sparse Markov and Independence Models", by M. Janzamin and A. Anandkumar, *Journal of Machine Learning Research*, 15:15491591, April 2014.

REFEREED CONFERENCE PUBLICATIONS

"Provable Tensor Methods for Learning Mixtures of Generalized Linear Models", by H. Sedghi, M. Janzamin, and A. Anandkumar, In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Cadiz, Spain, May 2016.

"FEAST at Play: Feature ExtrAction using Score function Tensors", by M. Janzamin, H. Sedghi, U.N. Niranjan, and A. Anandkumar In *NIPS Feature Extraction: Modern Questions and Challenges workshop*, Montreal, Canada, Dec 2015.

"Learning Overcomplete Latent Variable Models through Tensor Methods", by A. Anandkumar, R. Ge, and M. Janzamin*, In *Proceedings of the Conference on Learning Theory (COLT)*, Paris, France, July 2015.

"When are Overcomplete Topic Models Identifiable? Uniqueness of Tensor Tucker Decompositions with Structured Sparsity", by A. Anandkumar, D. Hsu, M. Janzamin*, and S. Kakade, In *Proceedings of the Neural Information Processing Systems (NIPS) Conference*, Lake Tahoe, Nevada, USA, Dec 2013.

"High-Dimensional Covariance Decomposition into Sparse Markov and Independence Domains", by M. Janzamin and A. Anandkumar, In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, June 2012.

"A Game-Theoretic Approach for Power Allocation in Bidirectional Cooperative Communication", by M. Janzamin, M. R. Pakravan, and H. Sedghi, In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Sydney, Australia, April 2010.

Preprints

"Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods", by M. Janzamin, H. Sedghi, and A. Anandkumar. *Submitted: Preprint available on arXiv:1506.08473*, June 2015.

"Score Function Features for Discriminative Learning: Matrix and Tensor Frameworks", by M. Janzamin, H. Sedghi, and A. Anandkumar, *Preprint available on arXiv:1412.2863*, Dec. 2014.

"Analyzing Tensor Power Method Dynamics in Overcomplete Regime", by A. Anandkumar, R. Ge, and M. Janzamin* *Accepted for publication in Journal of Machine Learning Research. available on arXiv:1411.1488.*

"Sample Complexity Analysis for Learning Overcomplete Latent Variable Models through Tensor Methods", by A. Anandkumar, R. Ge, and M. Janzamin*, *Preprint available on arXiv:1408.0553*, Aug. 2014.

"Guaranteed Non-Orthogonal Tensor Decomposition via Alternating Rank-1 Updates", by A. Anandkumar, R. Ge, and M. Janzamin*, *Preprint available on arXiv:1402.5180*, Feb. 2014.

ABSTRACT OF THE DISSERTATION

Non-convex Optimization in Machine Learning: Provable Guarantees Using Tensor Methods

By

Majid Janzamin

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2016

Professor Animashree Anandkumar, Chair

In the last decade, machine learning algorithms have been substantially developed and they have gained tremendous empirical success. But, there is limited theoretical understanding about this success. Most real learning problems can be formulated as *non-convex optimization* problems which are difficult to analyze due to the existence of several local optimal solutions. In this dissertation, we provide simple and efficient algorithms for learning some probabilistic models with provable guarantees on the performance of the algorithm. We particularly focus on analyzing *tensor methods* which entail non-convex optimization. Furthermore, our main focus is on challenging overcomplete models. Although many existing approaches for learning probabilistic models fail in the challenging overcomplete regime, we provide scalable algorithms for learning such models with low computational and statistical complexity.

In probabilistic modeling, the underlying structure which describes the observed variables can be represented by latent variables. In the *overcomplete* models, these hidden underlying structures are in a higher dimension compared to the dimension of observed variables. A wide range of applications such as speech and image are well-described by overcomplete models. In this dissertation, we propose and analyze *overcomplete tensor decomposition* methods and exploit them for learning several latent representations and latent variable models in the unsupervised setting. This include models such as mulitiview mixture model, Gaussian mixtures, Independent Component Analysis, and Sparse Coding (Dictionary Learning). Since latent variables are not observed, we also have the identifiability issue in latent variable modeling and characterizing latent representations. We

also propose sufficient conditions for identifiability of overcomplete topic models. In addition to unsupervised setting, we adapt the tensor techniques to supervised setting for learning neural networks and mixtures of generalized linear models.

Chapter 1

Introduction

Machine learning for handling large data sets is considered as one of the foremost grand challenges of the current time. It provides a principled approach for extracting useful information from data that can be exploited to carry out tasks such as prediction. In the past, machine learning algorithms have been mostly developed and inspired by practical motivations. Although, there also exist a significant number of studies on the theory of machine learning, for the most part of the literature there is a gap between the practical success of machine learning algorithms and understanding their fundamental theory. In this dissertation, we focus on the fundamental theory of machine learning. We work on statistical learning and probabilistic modeling techniques, wherein we fit a statistical model to the data. It is assumed that the data is generated from the proposed probabilistic model. This systematic approach of analyzing the machine learning models and algorithms is theoretically interesting, and furthermore, this provides us with useful intuitions and insights for designing new practical machine learning algorithms.

The focus of this dissertation is on developing learning algorithms which work well both in practice and theory. We provide simple and efficient algorithms for learning probabilistic models with provable guarantees on the performance of the algorithm. In particular, the main focus is on the challenging overcomplete regime. In probabilistic modeling, the underlying structure which describes the observed variables can be represented by latent variables. In *overcomplete* models,

these hidden underlying structures are in a higher dimension compared to the dimension of observed variables. A wide range of applications such as speech and image are well-described by overcomplete models. Designing and analyzing learning algorithms in the overcomplete regime is very challenging in both computational and statistical senses.

In addition, most real learning problems can be formulated as *non-convex optimization* problems which are difficult to analyze due to the existence of several local optimal solutions. In this dissertation, we specifically work on tensor methods which entail analyzing a class of non-convex optimization techniques. For these methods, we prove the convergence guarantees to global optima and useful local optimal solutions in the challenging overcomplete regime. We exploit and analyze these methods for learning several probabilistic models. We provide unsupervised techniques for learning latent variable models, and supervised methods for learning models such as neural networks.

Although many existing approaches for learning probabilistic models fail in the challenging overcomplete regime, we provide scalable algorithms for learning such models with low computational and statistical complexity. In addition, we also propose provable guarantees on the performance of the algorithm.

The rest of this chapter is organized as follows. We first summarize the organization of the dissertation. We then give a summary of main body of the dissertation including the problems analyzed in different chapters and our main contributions. Since tensors are significantly used and analyzed throughout the dissertation, we finally provide a section on tensor background.

1.1 Organization of the Dissertation

We provide the following results in this dissertation.

We propose provable guarantees for learning different latent representations and latent variable models in the overcomplete regime. This includes models such as the mulitiview mixture model, Gaussian mixtures, Independent Component Analysis, and Sparse Coding (Dictionary Learning).

The learning algorithm is based on tensor decomposition techniques. The observed moment is formed as a low order tensor (usually third or fourth order), and by decomposing the tensor to its rank-1 components we are able to learn the parameters of the model; see Section 3.3 which describes this connection. The analysis of tensor decomposition and the application to learning latent variable models are respectively provided in Chapters 2 and 3. The results in these two chapters are based on our publication in COLT'15; see Curriculum Vitae.

The above results on learning latent representations are primarily in the unsupervised setting. In Chapter 4, we adapt these techniques to supervised setting, where we exploit the cross-moment between output and a specific non-linear transformations of the input, and by decomposing that to rank-1 components, we learn the parameters of the model. In particular, we propose provable risk bounds for training neural networks with one hidden layer. The results in this chapter has some overlaps with our publication in AISTATS'16; see Curriculum Vitae.

In Chapter 5, we turn our attention to the identifiability question. Identifiability is concerned with the uniqueness of learning the parameters of the model, i.e., under what conditions the model can be uniquely learned? We propose identifiability results for overcomplete topic models. The results in this chapter are based on our publications in JMLR'15 and NIPS'13; see Curriculum Vitae.

The main body of the dissertation is self-contained such that all the results, discussions, proof outlines, and numerical experiments are included. For the sake of smoother reading experience, all the detailed proofs are postponed to the appendix.

1.2 Learning Overcomplete Latent Representations Using Tensor Methods

It is imperative to incorporate *latent or hidden variables* in any probabilistic modeling framework. In particular, incorporating latent variables is crucial for characterizing unobserved or hidden effects in statistical models. A wide range of applications such as document, speech and image modeling are well-characterized by incorporating latent variables. Latent variables have shown to

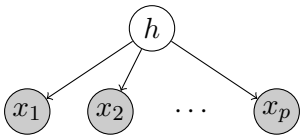


Figure 1.1: Multi-view mixtures model

be useful to provide a good explanation of the observed data, where they can capture the effect of hidden causes which are not directly observed. Learning these hidden factors is central to many applications, e.g., identifying latent diseases through observed symptoms, and identifying latent communities through observed social ties. Furthermore, latent representations are very useful in feature learning. Raw data is in general very complex and redundant and feature learning is about extracting efficient features from raw data. Learning efficient and useful features is very crucial for the good performance of learning task, e.g., the classification task that we do using the learned features. Analyzing latent variable models (LVMs) and latent representations involves two main problems: identifiability and learning. We focus on the learning part here, and later discuss the identifiability.

An interesting regime of latent representations is *overcomplete* setting, where the dimension of hidden space is more than the dimension of observed variables. Overcomplete representations are known to be more robust to noise, and can provide greater flexibility in modeling [119]. Overcomplete representations have been extensively employed, and are arguably critical in a number of applications such as speech and computer vision [40]. Many recent papers employ unsupervised estimation of overcomplete features for higher level tasks such as classification, e.g., [40, 57, 118, 71], and record significant gains in predictive accuracy over other approaches in a number of applications such as speech recognition and computer vision. However, since identifiability and learning are more challenging in a overcomplete regime, the *theoretical* understanding regarding learnability or identifiability of overcomplete representations is far more limited.

In the first part of this dissertation (Chapters 2 and 3), we provide tensor decomposition algorithms for learning several latent variable models and latent representations in the overcomplete regime. As one of the basic examples of latent variable models, consider the multiview linear mixtures model, where the observed variables (views) $x_l \in \mathbb{R}^d, l \in [p]$, are conditionally independent given

the k-categorical latent variable $h \in [k]$, and the conditional means are

$$\mathbb{E}[x_1|h] = a_h, \quad \mathbb{E}[x_2|h] = b_h, \quad \mathbb{E}[x_3|h] = c_h,$$

where $A := [a_1 \ a_2 \ \cdots \ a_k] \in \mathbb{R}^{d \times k}$ denotes the *factor matrix* and B, C are similarly defined; see Figure 1.1 for a graphical representation of this model. The goal of the learning problem is to recover the parameters of the model (factor matrices) A, B, and C given observations.

For this model, the third order observed moment is shown to have the form (see Anandkumar et al. [15])

$$\mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j,$$

where $\otimes$ denotes the outer product; see (1.4) for the precise definition. Hence, given the third order observed moment $\mathbb{E}[x_1 \otimes x_2 \otimes x_3]$, the unsupervised learning problem (recovering factor matrices A, B, and C) reduces to computing a tensor decomposition as stated above.

Regarding feature learning that we discussed its importance, dictionary learning is a popular model which represents the observations as a linear combination of latent dictionary elements. More precisely, for observation x, we have the representation

$$x = Ah,$$

where A denotes the dictionary matrix with dictionary elements as its columns and h denotes the coefficients. In the dictionary learning problem, both dictionary A and coefficients h are hidden or unknown and need to be estimated. Similar to the multiview mixture model, a variant of fourth order observed moment $\mathbb{E}[x \otimes x \otimes x \otimes x]$ is shown to have the columns of dictionary matrix A as its rank-1 components; see Section 3.3 for the details. Thus, the problem of learning dictionary elements reduces again to tensor decomposition. It is also very common to impose sparsity on the coefficient vector h, and then the problem is also called sparse coding.

In Chapter 2, we analyze the convergence properties of tensor decomposition in the overcomplete regime. The analysis of tensor decomposition is a non-convex problem for which there have been a good understanding of convergence guarantees, but these are mostly under strong assumptions. For instance, one of the main advantages of tensors compared to matrices is that tensors can have non-orthogonal decomposition which is not the case for matrices. This dramatically expands the class of probabilistic models that can be learned by the tensor decomposition methods including overcomplete models. But, on the other hand, the existing theoretical guarantees are mostly provided only for the orthogonal tensor decomposition. We provide new results on the local and global convergence guarantees of tensor decomposition methods in the non-orthogonal setting, and in particular in the overcomplete regime where the tensor rank is larger than the tensor dimension.

In Chapter 3, we provide the application of the tensor methods to learning many latent variable models and latent representations including multiview mixtures model, Gaussian mixtures, Independent Component Analysis, and Sparse Coding (Dictionary Learning). The main contribution of this chapter is analyzing new concentration bounds to argue the sample complexity results.

1.3 Guaranteed Training of Neural Networks

Neural networks have significantly improved predictive performance across multiple domains such as computer vision and speech recognition. Although there has been tremendous success of neural networks in practice, the theoretical understanding of this achievement is mostly limited. This is mainly because training a neural network is a highly non-convex problem, for which the analysis is difficult, and existing training methods can get stuck in local optima.

In Chapter 4, we propose a novel algorithm with provable guarantees for training neural networks with one hidden layer; see Figure 1.2 for a graphical representation of such neural network. This is the first analysis to effectively address both approximation and estimation problems in neural networks at the same time, and to propose computationally and statistically efficient methods. The approximation analysis deals with the problem of how good a neural network can approximate any arbitrary function, and the estimation analysis deals with the problem of how well the proposed

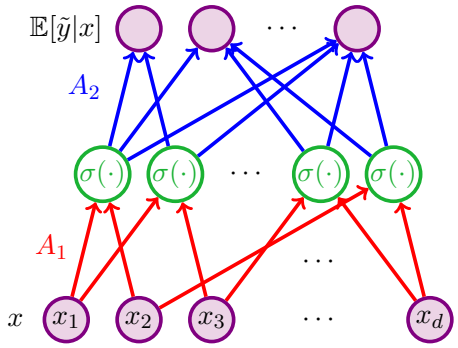


Figure 1.2: Graphical representation of a neural network, $\mathbb{E}[\tilde{y}|x] = A_2^\top \sigma(A_1^\top x + b_1) + b_2$.

algorithm can train the neural network given samples of that arbitrary function. We provide an algorithm which trains a neural network with bounded approximation and estimation errors (i.e., bounded risk), which is both computationally and statistically efficient. We incorporate tensor decomposition methods as the training algorithm. Tensor methods as multi-linear operators are highly parallelizable, and work very well in practice.

In Section 1.2, we discussed the the application of tensor methods for learning latent variable models and latent representations. Here, one of the main differences is that the problem of training neural network is in supervised setting, while learning latent variable models is performed in an unsupervised manner. Thus, one of the main questions that we answer is how to adapt these tensor methods to supervised learning, and in particular, training neural networks. To answer this, we first introduce a new transformation of the input which is basically new features extracted from the input. We refer to this new transformation as *s*core function of the input. These new extracted features enable us to formulate the problem of training neural networks as the tensor decomposition problem. More concretely, we show that the cross-moment between output and the score function of the input has information about the weight parameters of the neural network in its rank-1 components. Then we combine this novel formulation with the convergence properties of tensor decomposition to provably train the neural network.

1.4 Identifiability of Overcomplete Topic Models

In Section 1.2, we discussed the importance of latent variables in probabilistic modeling, and we provided learning results for several latent variable models in the overcomplete regime. Since latent variables are not observed, there exists an additional ambiguity in the model, where the same observations can be generated with different latent factors. Therefore, in addition to providing learning guarantees, we also need some sufficient conditions under which the model is identifiable, i.e., when the model parameters and latent variables can be uniquely identified given some observed statistics. In the *overcomplete* case, where the latent dimensionality is greater than the observed dimensionality, it is even more challenging to provide model identifiability result. Intuitively, we are observing variables in a lower dimension comparing to the dimension of hidden factors which makes the model non-identifiable in general.

Latent variable modeling has been very popular in topic modeling application, where the words in the documents are observed, while the topics of documents are hidden. More concretely, let x denote an observed variable (word), and h denote the hidden topic proportion vector. In probabilistic topic modeling, the following linear mapping from latent variables (topics) to observed variables (words) holds such that

$$\mathbb{E}[x|h] = Ah,$$

where A incorporates the influence of hidden factors on observed variables, called the *topic-word matrix*.

In Chapter 5, we provide the identifiability results for learning the parameters of overcomplete topic models under moment-based observations. Note that these overcomplete topic models are more interesting and useful in speech and image applications, and less attractive for text applications. Given higher order moments of observed variables, we provide the identifiability result under both deterministic and random assumptions on the topic-word matrix A. This involves leveraging combinatorial conditions on the graph which represents the topic-word relations in the topic modeling, where the main identifiability condition imposes sparsity on matrix A. Incorporating this

sparsity structure on A, the model parameter A and latent variables are estimated using convex optimization techniques for sparse recovery.

Furthermore, the identifiability result implies uniqueness of a class of *tensor decompositions* with structured sparsity. These class of tensor decompositions are a special case of Tucker, but more general than CP decomposition.

1.5 Tensor Overview

In this section, we first introduce the tensor background and tensor notations. Then, we elaborate and motivate the necessity of tensors over matrices.

1.5.1 Tensor Background and Notations

A real-valued *p-th order tensor* $T \in \bigotimes_{i=1}^{p} \mathbb{R}^{d_i}$ is a member of the outer product of Euclidean spaces $\mathbb{R}^{d_i}$, $i \in [p]$. For convenience, we restrict to the case where $d_1 = d_2 = \cdots = d_p = d$, and simply write $T \in \bigotimes^{p} \mathbb{R}^{d}$. As is the case for vectors (where $p = 1$) and matrices (where $p = 2$), we may identify a p-th order tensor with the p-way array of real numbers $[T_{i_1,i_2,\ldots,i_p} : i_1, i_2, \ldots, i_p \in [d]]$, where $T_{i_1,i_2,\ldots,i_p}$ is the $(i_1, i_2, \ldots, i_p)$-th coordinate of T with respect to a canonical basis.

Tensor modes, fibers and slices: The different dimensions of the tensor are referred to as *modes*. For instance, for a matrix, the first mode refers to columns and the second mode refers to rows. In addition, *fibers* are higher order analogues of matrix rows and columns. A fiber is obtained by fixing all but one of the indices of the tensor (and is arranged as a column vector). For instance, for a matrix, its mode-1 fiber is any matrix column while a mode-2 fiber is any row. For a third order tensor $T \in \mathbb{R}^{d \times d \times d}$, the mode-1 fiber is given by $T(:, j, l)$, mode-2 by $T(i, :, l)$ and mode-3 by $T(i, j, :)$. Similarly, *slices* are obtained by fixing all but two of the indices of the tensor. For example, for the third order tensor T, the slices along 3rd mode are given by $T(:, :, l)$.

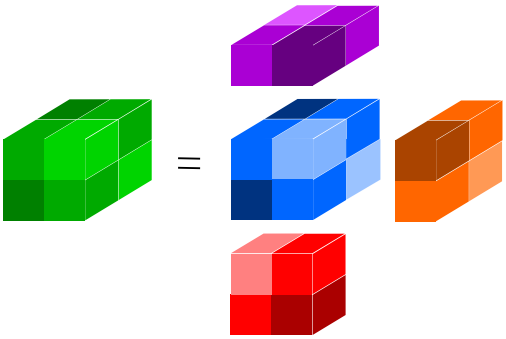


Figure 1.3: Tensor as a multilinear transformation and representation of Tucker decomposition of a symmetric 3rd order tensor $T = \sum_{i_1,i_2,i_3 \in [k]} S_{i_1,i_2,i_3} \cdot a_{i_1} \otimes b_{i_2} \otimes c_{i_3} = S(A^\top, B^\top, C^\top)$

Tensor matricization: For $r \in \{1,2,3\}$, the mode-r matricization of a third order tensor $T \in \mathbb{R}^{d\times d\times d}$, denoted by $\mathrm{mat}(T,r) \in \mathbb{R}^{d\times d^2}$, consists of all mode-$r$ fibers arranged as column vectors. For instance, the matricized version along first mode denoted by $M \in \mathbb{R}^{d\times d^2}$ is defined such that

$$T(i,j,l) = M(i, l+(j-1)d), \quad i,j,l \in [d]. \tag{1.1}$$

Multilinear transformation: We view a tensor $T \in \mathbb{R}^{d\times d\times d}$ as a multilinear form. Consider matrices $A, B, C \in \mathbb{R}^{d\times k}$. Then tensor $T(A,B,C) \in \mathbb{R}^{k\times k\times k}$ is defined as

$$T(A,B,C)_{j_1,j_2,j_3} := \sum_{i_1,i_2,i_3 \in [d]} T_{i_1,i_2,i_3} \cdot A(i_1,j_1) \cdot B(i_2,j_2) \cdot C(i_3,j_3). \tag{1.2}$$

See Figure 1.3 for a graphical representation of multilinear form. In particular, for vectors $u, v, w \in \mathbb{R}^d$, we have [1]

$$T(I,v,w) = \sum_{j,l \in [d]} v_j w_l T(:,j,l) \ \in \mathbb{R}^d, \tag{1.3}$$

which is a multilinear combination of the tensor mode-1 fibers. Similarly $T(u,v,w) \in \mathbb{R}$ is a multilinear combination of the tensor entries, and $T(I,I,w) \in \mathbb{R}^{d\times d}$ is a linear combination of the tensor slices.

[1] Compare with the matrix case where for $M \in \mathbb{R}^{d\times d}$, we have $M(I,u) = Mu := \sum_{j\in[d]} u_j M(:,j) \in \mathbb{R}^d$.

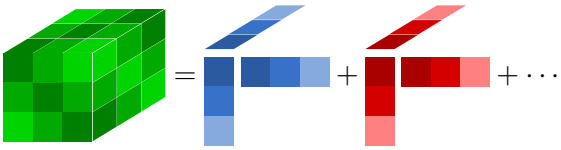


Figure 1.4: CP decomposition of a symmetric 3rd order tensor $T = \sum_i a_i \otimes a_i \otimes a_i$

Rank-1 tensor: A 3rd order tensor $T \in \mathbb{R}^{d \times d \times d}$ is said to be rank-1 if it can be written in the form

$$T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l), \tag{1.4}$$

where notation $\otimes$ represents the *outer product* and $a \in \mathbb{R}^d$, $b \in \mathbb{R}^d$, $c \in \mathbb{R}^d$ are unit vectors (without loss of generality).

Tensor CP decomposition and rank: A tensor $T \in \mathbb{R}^{d \times d \times d}$ is said to have a CP (CANDECOMP/PARAFAC) rank $k \geq 1$ if it can be written as the sum of k rank-1 tensors

$$T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad w_i \in \mathbb{R},\ a_i, b_i, c_i \in \mathbb{R}^d. \tag{1.5}$$

See Figure 1.4 for a graphical representation of CP decomposition for a symmetric 3rd order tensor. This decomposition is also closely related to the multilinear form. In particular, given T in (1.5) for vectors $\widehat{a}, \widehat{b}, \widehat{c} \in \mathbb{R}^d$, we have

$$T(\widehat{a}, \widehat{b}, \widehat{c}) = \sum_{i \in [k]} w_i \langle a_i, \widehat{a} \rangle \langle b_i, \widehat{b} \rangle \langle c_i, \widehat{c} \rangle.$$

Consider the decomposition in equation (1.5), denote matrix $A := [a_1\ a_2\ \cdots\ a_k] \in \mathbb{R}^{d \times k}$, and similarly B and C. Without loss of generality, we assume that the matrices have normalized columns (in 2-norm), since we can always rescale them, and adjust the weights w_i appropriately.

Tensor Tucker decomposition: A tensor $T \in \mathbb{R}^{d \times d \times d}$ is said to have a Tucker decomposition or Tucker representation when given core tensor $S \in \mathbb{R}^{k \times k \times k}$ and factor matrices $A, B, C \in \mathbb{R}^{d \times k}$, it

can be written as

$$T = \sum_{i_1 \in [k]} \sum_{i_2 \in [k]} \sum_{i_3 \in [k]} S_{i_1,i_2,i_3} \cdot a_{i_1} \otimes b_{i_2} \otimes c_{i_3}. \tag{1.6}$$

See Figure 1.3 for a graphical representation. Note that this is directly related to the multilinear from defined in (1.2) such that the R.H.S. of above equation is $S(A^\top, B^\top, C^\top)$. Note that the CP decomposition is a special case of the Tucker decomposition when the core tensor S is square and diagonal.

Norms: Throughout, $\|v\| := (\sum_i v_i^2)^{1/2}$ denotes the Euclidean (ℓ_2) norm of a vector v, and $\|M\|$ denotes the spectral (operator) norm of a matrix M. Furthermore, $\|T\|$ and $\|T\|_F$ denote the spectral (operator) norm and the Frobenius norm of a tensor, respectively. In particular, for a 3rd order tensor, we have

$$\|T\| := \sup_{\|u\|=\|v\|=\|w\|=1} |T(u,v,w)|, \quad \|T\|_F := \sqrt{\sum_{i,j,l \in [d]} T_{i,j,l}^2}.$$

We finish this section by introducing the asymptotic notations.

Asymptotic notations: Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. While the standard asymptotic notation is to write $f(d) = O(g(d))$ and $g(d) = \Omega(f(d))$, we sometimes use $f(d) \leq O(g(d))$ and $g(d) \geq \Omega(f(d))$ for additional clarity. We also use the asymptotic notation $f(d) = \tilde{O}(g(d))$ if and only if $f(d) \leq \alpha g(d)$ for all $d \geq d_0$, for some $d_0 > 0$ and $\alpha = \text{polylog}(d)$, i.e., $\tilde{O}$ hides polylog factors.

1.5.2 Going Beyond Matrices

Tensor decomposition is by itself a theoretically interesting problem and is in particular very challenging in the overcomplete regime. In this dissertation, we analyze the convergence guarantees of a specific tensor decomposition algorithm to recover the true rank-1 components of the tensor. Tensor decomposition has also applications in many different areas such as chemometrics [26],

neuroscience [126], telecommunications [144], data mining [1], image compression and classification [142], and so on; see survey paper by Kolda and Bader [108] for more references. In the earlier parts of this chapter, we stated that tensor decomposition is also useful in machine learning applications, and in particular, for learning latent variable models and latent representations. A few examples such as learning multiview mixture model and dictionary learning problems were discussed in Section 1.2; more examples are provided in Section 3.3. Tensors are basically generalization of matrices to higher order objects, and beyond that tensors are multilinear operators as defined in (1.2). Going back to the example of learning multiview mixture model in Section 1.2, we can form the second order observed moment which has the form

$$\mathbb{E}[x_1 \otimes x_2] = \sum_{j \in [k]} w_j a_j \otimes b_j.$$

So, one fundamental question is why do we need to go to tensors and matrices are not enough for our learning task? We answer this question to motivate and justify the application of tensors.

We require at least a third order tensor to learn the parameters of the latent representation or latent variable models for the following reasons: while a matrix decomposition is only identifiable up to orthogonal components, tensors can have identifiable non-orthogonal components. In general, it is not realistic to assume that the parameters are orthogonal, and hence, we require tensors to learn the parameters. Moreover, through tensors, we can learn overcomplete models, where the hidden dimension can exceed the input dimension. Note that matrix factorization methods are unable to learn overcomplete models, since the rank of the matrix cannot exceed its dimensions. Thus, it is critical to incorporate tensors for such models.

Regarding the identifiability, the sufficient conditions for uniqueness of tensor decomposition is formulated by Kruskal [111, 112]. Let the Kruskal rank or krank of a matrix A denoted by $\text{krank}(A)$ be the maximum number r such that every subset of r columns of A is linearly independent. Kruskal [111, 112] showed that the decomposition in (1.5) is unique if

$$\text{krank}(A) + \text{krank}(B) + \text{krank}(C) \geq 2k + 2,$$

which is a milder condition compared to matrix decomposition. For instance, for a rank-2 decomposition, this reduces to having all factor matrices A, B and C being full column rank, while in the matrix case we need the stronger condition of orthogonality.

Chapter 2

Overcomplete CP Tensor Decomposition

CANDECOMP/PARAFAC (CP) decomposition of a symmetric tensor $T \in \mathbb{R}^{d \times d \times d}$ is the process of decomposing it into a succinct sum of rank-one tensors, given by

$$T = \sum_{j \in [k]} \lambda_j a_j \otimes a_j \otimes a_j, \quad \lambda_j \in \mathbb{R},\ a_j \in \mathbb{R}^d, \tag{2.1}$$

where $\otimes$ denotes the outer product; see (1.4) for the precise definition. The minimum k for which the tensor can be decomposed in the above form is called the (symmetric) tensor rank.

In this chapter, we provide local and global convergence guarantees for recovering CP (Candecomp/Parafac) tensor decomposition in the overcomplete regime where the tensor CP rank is larger than the input dimension. Finding the CP decomposition of an overcomplete tensor is NP-hard in general. We analyze the simple tensor power iteration, and provide the convergence guarantees under two settings of incoherent and random tensor components. We also propose tight perturbation analysis given noisy tensor for both settings.

Given incoherent tensor components, local convergence guarantees are established for third order tensors of rank k in d dimensions, when $k = o\big(d^{1.5}\big)$. Thus, we can recover overcomplete tensor

decomposition where the tensor rank k is larger than the dimension d. We also strengthen the results to global convergence guarantees under stricter rank condition $k \leq \beta d$ (for arbitrary constant $\beta > 1$) through a simple initialization procedure where the algorithm is initialized by top singular vectors of random tensor slices. Furthermore, the approximate local convergence guarantees for p-th order tensors are also provided under rank condition $k = o\big(d^{p/2}\big)$.

Next, we strengthen the local convergence analysis given random tensor components for third order tensors. We show that the simple power iteration recovers the components with bounded error under mild initialization conditions. These initialization conditions are much more relaxed compared to the conditions under incoherent tensor components.

CP (Candecomp/Parafac) tensor decomposition became popular in the psychometrics community by the works of Carroll and Chang [49], Harshman [86]. Later, researchers also applied these techniques to several different research areas including chemometrics [26], neuroscience [126], telecommunications [144], data mining [1], image compression and classification [142], and many other applications; see survey paper by Kolda and Bader [108] for more references. They have also been recently popular for unsupervised learning of a wide range of latent variable models such as independent component analysis (ICA) [69, 70], topic models, Gaussian mixtures, hidden Markov models [15], network community models [12], and so on.

Tensor power iteration is a simple, popular and efficient method for recovering the tensor rank-one components a_j's in (2.1). The tensor power iteration is given by

$$x \leftarrow \frac{T(I, x, x)}{\|T(I, x, x)\|}, \tag{2.2}$$

where

$$T(I, x, x) := \sum_{j,l \in [d]} x_j x_l T(:, j, l) \in \mathbb{R}^d$$

is a *multilinear* combination of tensor *fibers*, and $\|\cdot\|$ is the ℓ_2 norm operator. See Section 1.5.1 for an overview of tensor notations and preliminaries.

The tensor power iteration is a generalization of matrix power iteration: for matrix $M \in \mathbb{R}^{d \times d}$, the power iteration is given by $x \leftarrow Mx/\|Mx\|$. Dynamics and convergence properties of matrix power iterations are well understood [91]. On the other hand, a theoretical understanding of tensor power iterations is much more limited. Tensor power iteration can be viewed as a *gradient descent* step (with infinite step size), corresponding to the optimization problem

$$\max_{x \in \mathcal{S}^{d-1}} |T(x, x, x)|,$$

where $T(x, x, x) = \sum_{i,j,l \in [d]} x_i x_j x_l T_{ijl} \in \mathbb{R}$ is a combination of entries of tensor T. This optimization problem is *non-convex*, and has multiple local optima. Unlike the matrix case, where the number of isolated stationary points of power iteration is at most the dimension (given by eigenvectors corresponding to unique eigenvalues), in the tensor case, the number of stationary points is, in fact, exponential in the input dimension [50]. This makes the analysis of tensor power iteration far more challenging.

Despite the above challenges, many advances have been made in understanding the tensor power iterations in specific regimes. When the components a_j's are orthogonal to one another, it is known that there are no spurious local optima for tensor power iterations, and the only stable fixed points correspond to the true a_j's [160, 18]. Any tensor with linearly independent components a_j's can be orthogonalized, via an invertible transformation (whitening) and thus, its components can be recovered efficiently. A careful perturbation analysis in this setting was carried out in Anandkumar et al. [18].

While having efficient guarantees, the above procedure for tensor decomposition suffers from a number of theoretical and practical limitations. For instance, in practice, the learning performance is especially sensitive to whitening [117]. Moreover, whitening is computationally the most expensive step in deployments [97], and it can suffer from numerical instability in high-dimensions due to ill-conditioning. Lastly, the above approach is unable to learn *overcomplete representations* (this is the case when the tensor rank is larger than the dimension) due to the orthogonality constraint, which is especially limiting, given the recent popularity of overcomplete feature learning in many domains [40, 119]. Such overcomplete tensors cannot be orthogonalized and finding guaranteed

decomposition is a challenging open problem. It is known that finding CP tensor decomposition is NP-hard [88]. In this thesis, we make significant headway in showing that the simple power iterations can recover the components in the overcomplete regime under a set of mild conditions on the components a_j's.

In this chapter, we provide two main results for overcomplete tensor decomposition. In the first and main part, the presence of *incoherent* tensor components is assumed, which can be viewed as a *soft-orthogonality* constraint. Incoherent representations have been extensively considered in literature in a number of contexts, e.g., compressed sensing [74] and sparse coding [27, 3]. Incoherent representations provide flexible modeling, can handle overcomplete signals, and are robust to noise [119]. Moreover, in the application to learning latent variable models, when the parameters of the model are *generic* or when we have randomly constructed (multiview) features [123], the moment tensors have incoherent components, as assumed here. In this work, we establish that incoherence leads to efficient guarantees for tensor decomposition. The guarantees also include a tight perturbation analysis. In the second part, we provide stronger tensor decomposition guarantees under *random* rank-1 components. Note that random components are also incoherent, but the reverse is not true.

Note that overcomplete tensors arise in many machine learning applications such as moments of many latent variable models, e.g., multiview mixtures, independent component Analysis (ICA), and sparse coding models, where the number of hidden variables exceeds the input dimensions [16]. Overcomplete models often have impressive empirical performance [58], and can provide greater flexibility in modeling, and are more robust to noise [119]. By studying algorithms for overcomplete tensor decomposition, we expand the class of models that can be learnt efficiently using simple spectral methods such as tensor power iterations. Note there are other algorithms for decomposing overcomplete tensors [69, 80, 41], but they all require tensors of at least 4-th order and require large computational complexity. Ge and Ma [77] works for 3rd order tensor but requires quasi-polynomial time. The main contribution of this thesis is an analysis for the practical power method in the overcomplete regime.

In next chapter, we apply the tensor decomposition guarantees of this chapter to various learning settings, and derive sample complexity bounds through novel covering arguments.

2.1 Summary of results

Consider an asymmetric tensor $T \in \mathbb{R}^{d\times d\times d}$ decomposed to rank-1 components as

$$T = \sum_{i\in[k]} w_i \cdot a_i \otimes b_i \otimes c_i, \quad w_i \in \mathbb{R},\ a_i, b_i, c_i \in \mathbb{R}^d. \tag{2.3}$$

The goal is to recover its rank-1 components $\{(a_i, b_i, c_i), i \in [k]\}$. Here, $\otimes$ denotes the tensor outer product; see Section 1.5.1 for the details of tensor notations and tensor rank.

In this chapter, we propose and analyze an algorithm for non-orthogonal CP (Candecomp/Parafac) tensor decomposition; see Figure 2.1 for the details of the algorithm. The main step of the algorithm is a simple alternating rank-1 update which is the alternating version of the tensor power iteration adapted for asymmetric tensors. In each iteration, one of the tensor modes is updated by projecting the other modes along their estimated directions, and the process is alternated between all the modes of the tensor; see (2.6) for this update and compare it with (2.2) which is the symmetric tensor power iteration where there is no need for alternating between different modes.

We provide the convergence results under two different settings. First, given incoherent rank-1 components, and then given random rank-1 components. In the second part, by imposing stronger assumption of randomness, we are able to provide stronger convergence guarantees. We propose these results in the following two subsections.

2.1.1 Guarantees under incoherent components

An overview of our tensor decomposition algorithm is provided in Figure 2.1. The main step of our tensor decomposition algorithm is *alternating asymmetric tensor power update*; see(2.6) for this update. We provide robust analysis of the algorithm leading to local and global convergence guarantees when the input tensor is noisy.[1] Our analysis emphasizes on the challenging *overcomplete* regime where the tensor rank is larger than the dimension, i.e., $k > d$.

[1]Note that in the learning applications, we form the empirical moments as the input tensor which is noisy.

We require natural deterministic conditions on the tensor components to argue the convergence guarantees; see Appendix A.2 for the details. All of these conditions are satisfied if the true rank-1 components of the tensor are uniformly i.i.d. drawn from the unit d-dimensional sphere $\mathcal{S}^{d-1}$. Among the deterministic assumptions, the most important one is the *incoherence* condition which imposes a soft-orthogonality constraint between different rank-1 components of the tensor.

In the *local convergence* guarantee, we analyze the convergence properties of the algorithm assuming we have good initialization vectors for the non-convex tensor decomposition algorithm.

Theorem 2.1 (Local convergence guarantee of the tensor decomposition algorithm)**.** *Consider noisy rank-k tensor* $\widehat{T} = T + \Psi$ *as the input to the tensor decomposition algorithm, where* $T = \sum_{i\in[k]} w_i \cdot a_i \otimes b_i \otimes c_i$*, and* $\psi := \|\Psi\| \leq w_{\min}/6$*. Let the rank condition* $k \leq o\left(d^{1.5}\right)$ *is satisfied. Assuming we have good initialization vectors (which have constant correlation with the true components), then the algorithm outputs estimates* $\widehat{A} := [\widehat{a}_1 \cdots \widehat{a}_k] \in \mathbb{R}^{d\times k}$ *and* $\widehat{w} := [\widehat{w}_1 \cdots \widehat{w}_k]^\top \in \mathbb{R}^k$*, satisfying w.h.p.*

$$\left\|\widehat{A} - A\right\|_F \leq \tilde{O}\left(\frac{\sqrt{k}\cdot\psi}{w_{\min}}\right), \quad \|\widehat{w} - w\| \leq \tilde{O}\left(\sqrt{k}\cdot\psi\right).$$

Same error bounds hold for other factor matrices $B := [b_1 \cdots b_k]$ *and* $C := [c_1 \cdots c_k]$*. The number of iterations is* $N = \Theta\left(\log(1/\hat{\epsilon}_R)\right)$ *where* $\hat{\epsilon}_R := \min\{\psi/w_{\min}, \tilde{O}\big(\sqrt{k}/d\big)\}$*.*

Thus, we can decompose the tensor in the highly overcomplete regime $k \leq o\left(d^{1.5}\right)$. The $\sqrt{k}$ factor in the bound is from the fact that the final recovery guarantee is on the Frobenius norm of the whole factor matrix A. In the following, we provide stronger column-wise guarantees (where there is no $\sqrt{k}$ factor) with the expense of having an additional residual error term. Our algorithm includes two main update steps including tensor power iteration in (2.6) and residual error removal in (2.10). The guarantee for the first step — tensor power iteration — is

Lemma 2.1 (Local convergence guarantee of the tensor power updates)**.** *Consider the same settings as in Theorem 2.1. Then, the outputs of tensor power iteration steps in our algorithm satisfy w.h.p.*[2]

$$\min_{z\in\{-1,1\}} \|z\widehat{a}_j - a_j\| \leq \tilde{O}\left(\frac{\psi}{w_{\min}}\right) + \tilde{O}\left(\frac{\sqrt{k}}{d}\right), \quad |\widehat{w}_j - w_j| \leq \tilde{O}\left(\psi\right) + \tilde{O}\left(w_{\max}\frac{\sqrt{k}}{d}\right), \quad j \in [k].$$

Same error bounds hold for other factor matrices B and C.

The above result provides guarantees with the additional residual error $\tilde{O}\left(\frac{\sqrt{k}}{d}\right)$, but we believe this result also has independent importance for the following reasons. The above result provides column-wise guarantees which is stronger than the guarantees on the whole factor matrix in Theorem 2.1. Furthermore, we can only have recovery guarantees for a subset of rank-1 components of the tensor (the ones for which we have good initializations) without worrying about the rest of components. Finally, in the high-dimensional regime (large d), the residual error term goes to zero.

For the *global convergence* guarantee, we obtain good initialization vectors by performing a rank-1 SVD on the random slices of the moment tensor.

Theorem 2.2 (Global convergence guarantee of the tensor decomposition algorithm)**.** *Consider the same input tensor to the algorithm as in Theorem 2.1 with noise bound $\psi := \|\Psi\| \leq \tilde{O}\big(w_{\min}/\sqrt{d}\big)$. Let $k \leq \beta d$ (for arbitrary constant $\beta > 1$), and the initialization is performed by SVD-based method in Procedure 2 (in the Appendix) using a polynomial number of initializations scaled as k^{β^2}. Then, the same guarantees as in Theorem 2.1 hold.*

Note that the argument in Lemma 2.2 can be similarly adapted leading to global convergence guarantee of the tensor power iteration step.

For 4th and higher order tensors, same techniques can be exploited to argue similar results.

Overview of techniques: Greedy or rank-1 updates are perhaps the most natural procedure for CP tensor decomposition. For orthogonal tensors, they lead to guaranteed recovery [160]. However, when the tensor is non-orthogonal, greedy procedure is not optimal in general [107]. Finding tensor

[2]Note that recovery of components is up to sign. This is because a third order tensor is unchanged if the sign along one of the modes is fixed and the signs along the other two modes are flipped.

decomposition in general is NP-hard [88]. We circumvent this obstacle by limiting ourselves to tensors with incoherent components. We exploit incoherence to prove error contraction under each step of the alternating update procedure with an approximation error, which is decaying, when $k = o(d^{1.5})$. To this end, we require tools from random matrix theory, bounds on $2 \to p$ norm for random matrices [81, 2] for some $p < 3$, and matrix perturbation results to provide tight bounds on error contraction.

2.1.2 Guarantees under random components

In this section, we provide the summary of results under stronger random assumption on the rank-1 components. This result is for third order tensors. We assume that the tensor components a_j's are randomly drawn from the unit sphere. Since general tensor decomposition is challenging in the overcomplete regime, we argue that this is a natural first step to consider for tractable recovery.

We characterize the basin of attraction for the local optima near the rank-one components a_j's. We show that under mild initialization condition, there is fast convergence to these local optima in $O(\log \log d)$ iterations (i.e., quadratic convergence as opposed to linear convergence in case of matrices). This result is the core technical analysis of this part stated in the following theorem.

Theorem 2.3 (Dynamics of tensor power iteration)**.** *Consider tensor $\widehat{T} = T + E$ such that exact tensor T has rank-k decomposition in* (2.1) *with rank-one components $a_j \in \mathbb{R}^d, j \in [k]$ being uniformly i.i.d. drawn from the unit d-dimensional sphere, and the ratio of maximum and minimum (in absolute value) weights λ_j's being constant. In addition, suppose the perturbation tensor E has bounded norm as*

$$\|E\| \leq \epsilon \frac{\sqrt{k}}{d}, \quad \text{where} \quad \epsilon < o\left(\frac{\sqrt{k}}{d}\right). \tag{2.4}$$

Let tensor rank $k = o(d^{1.5})$, and the unit-norm initial vector $x^{(1)}$ satisfy the correlation bound

$$|\langle x^{(1)}, a_j \rangle| \geq d^{\beta} \frac{\sqrt{k}}{d}, \tag{2.5}$$

w.r.t. some true component $a_j, j \in [k]$*, for some constant* $\beta > 0$*. After* $N = \Theta(\log \log d)$ *iterations, the tensor power iteration in* (2.2) *outputs a vector having w.h.p. a constant correlation with the true component* a_j *as* $|\langle x^{(N+1)}, a_j \rangle| \geq 1 - \gamma$*, for any fixed constant* $\gamma > 0$*.*

As a corollary, this result can be used for learning latent variable models such as multiview mixtures. We show that the above initialization condition is satisfied using a sample with mild signal-to-noise ratio; see Section 3.7 for more details on this.

The above result is a significant improvement over the analysis in Theorem 2.1 for overcomplete tensor decomposition. In Theorem 2.1, it is required for the initialization vectors to have a constant amount of correlation with the true a_j's. However, obtaining such strong initializations is usually not realistic in practice. On the other hand, the initialization condition in (2.5) is mild, and decaying even when the rank k is significantly larger than dimension d; up to $k = o(d^{1.5})$. In learning the mixture model, such initialization vectors can be obtained as samples from the mixture model, even when there is a large amount of noise. Given this improvement, we combine our analyses in Theorems 2.3 and 2.1, proving that the model parameters can be recovered consistently.

Overview of proof techniques: A detailed proof outline for Theorem 2.3 is provided in Section 2.6. Under the random assumption, it is not hard to show that the first iteration of tensor power update makes progress. However, after the first iteration, the input vector and the tensor components are no longer *independent* of each other. Therefore, we cannot directly repeat the same argument for the second step.

How do we analyze the second step even though the vector and tensor components are correlated? The main intuition is to characterize the dependency between the vector and the tensor components, and show that there is still enough randomness left for us to repeat the argument. This idea was inspired by the analysis of Approximate Message Passing (AMP) algorithms [39]. However, our analysis here is very different in several key aspects: 1) In approximate message passing, typically the analysis works in the *large system limit*, where the number of iterations is fixed and the dimension goes to infinity. Here we can handle a superconstant number of iterations $O(\log \log d)$,

even for finite d; 2) Usually k is assumed to be a constant factor times d in the AMP-like analysis, while here we allow them to be polynomially related.

2.2 Related Works

CANDECOMP tensor decomposition [49], also known as PARAFAC decomposition [86, 87] is a classical definition for tensor decomposition with many applications. The most commonly used algorithm for CP decomposition is Alternating Least Squares (ALS) [62], which has no convergence guarantees in general. Kolda [107] and Zhang and Golub [160] analyze the greedy or the rank-1 updates in the orthogonal setting. In the noisy setting, Anandkumar et al. [15] analyze deflation procedure for orthogonal decomposition, and Song et al. [147] extend the analysis to the nonparametric setting. For the non-orthogonal tensors, a common strategy is to first apply a procedure called *whitening* to convert it to the orthogonal case. But as discussed earlier, the whitening procedure can lead to poor performance and bad sample complexity. Moreover, it requires the tensor factors to have full column rank, which rules out overcomplete tensors.

Learning overcomplete tensors is challenging, and they may not even be identifiable in general. Kruskal [111, 112] provided an identifiability result based on the *Kruskal* rank of the factor matrices of the tensor. Domanov and De Lathauwer [72, 73] also provide uniqueness conditions based on Khatri-Rao products of compound matrices of factor matrices. However, these results is limiting since it requires $k = O(d)$, where k is the tensor rank and d is the dimension. The FOOBI procedure by De Lathauwer et al. [69] overcomes this limitation by assuming *generic* factors, and shows that a polynomial-time procedure can recover the tensor components when $k = O(d^2)$ for fourth order tensors. However, the procedure does not work for third-order overcomplete tensors, and has no polynomial sample complexity bounds. Simple procedures can recover overcomplete tensors for higher order tensors (five or higher). For instance, for the fifth order tensor, when $k = O(d^2)$, we can utilize random slices along a mode of the tensor, and perform simultaneous diagonalization on the matricized versions. Note that this procedure cannot handle the same level of overcompleteness as FOOBI, since an additional dimension is required for obtaining two (or more) fourth order tensor

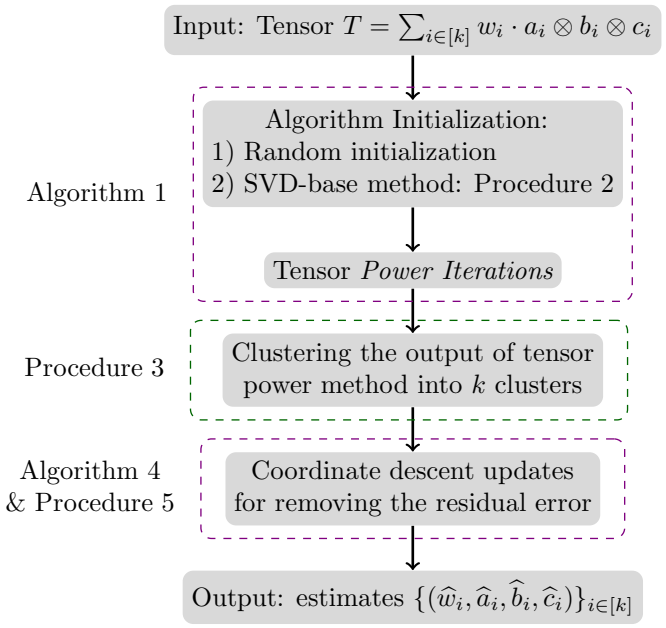


Figure 2.1: Overview of tensor decomposition algorithm.

slices. The simultaneous diagonalization procedure entails careful perturbation analysis, carried out by [80, 41]. In addition, Goyal et al. [80] provide stronger results for independent components analysis (ICA), where the tensor slices can be obtained in the Fourier domain.

There are other recent works which can learn overcomplete models, but under different settings than the ones considered in this work. For instance, Arora et al. [27], Agarwal et al. [3] provide guarantees for the sparse coding problem.

The algorithm employed here falls under the general framework of alternating minimization. There are many recent works which provide guarantees on local/global convergence for alternating minimization, e.g., for matrix completion [101, 84], phase retrieval [130] and sparse coding [3]. However, the techniques in this work are significantly different, since they involve tensors, while the previous works only required matrix analysis.

2.3 Tensor Decomposition Algorithm

In this section, we introduce the alternating tensor decomposition algorithm, and the guarantees are provided in Section 2.4. A summary of results are also provided in Section 2.1. The goal of tensor decomposition algorithm is to recover the rank-1 components of tensor; see (1.5) for the notion of tensor rank. Figure 2.1 depicts an overview of our tensor decomposition method where the corresponding algorithms and procedures are also specified. Our algorithm includes two main steps as 1) alternating tensor power iteration, and 2) coordinate descent iteration for removing the residual error. The former one is performed in Algorithm 1 (see equation (2.6)), and the latter one is done in Algorithm 4 (see equation (2.10)). We now describe these steps of the algorithm in more details as well as providing the auxiliary procedures required to complete the algorithm.

2.3.1 Tensor power iteration in Algorithm 1

The main step of the algorithm is tensor power iteration which basically performs alternating *asymmetric power updates*[3] on different modes of the tensor as

$$\widehat{a}^{(t+1)} = \frac{T\left(I, \widehat{b}^{(t)}, \widehat{c}^{(t)}\right)}{\left\|T\left(I, \widehat{b}^{(t)}, \widehat{c}^{(t)}\right)\right\|}, \quad \widehat{b}^{(t+1)} = \frac{T\left(\widehat{a}^{(t)}, I, \widehat{c}^{(t)}\right)}{\left\|T\left(\widehat{a}^{(t)}, I, \widehat{c}^{(t)}\right)\right\|}, \quad \widehat{c}^{(t+1)} = \frac{T\left(\widehat{a}^{(t)}, \widehat{b}^{(t)}, I\right)}{\left\|T\left(\widehat{a}^{(t)}, \widehat{b}^{(t)}, I\right)\right\|}, \tag{2.6}$$

where $\{\widehat{a}^{(t)}, \widehat{b}^{(t)}, \widehat{c}^{(t)}\}$ denotes estimate in the t-th iteration. Recall that for vectors $v, w \in \mathbb{R}^d$, the multilinear form $T(I, v, w) \in \mathbb{R}^d$ used in the above update formula is defined in (1.3), where $T(I, v, w)$ is a multilinear combination of the tensor mode-1 fibers. Notice that the updates alternate among different modes of the tensor which can be viewed as a rank-1 form of the standard Alternating Least Squares (ALS) method. We later discuss this relation in more details.

Optimization viewpoint: Consider the problem of best rank-1 approximation of tensor T as

$$\min_{\substack{a,b,c \in \mathcal{S}^{d-1} \\ w \in \mathbb{R}}} \|T - w \cdot a \otimes b \otimes c\|_F, \tag{2.7}$$

[3] This is exactly the generalization of asymmetric matrix power update to 3rd order tensors.

Algorithm 1 Tensor decomposition via alternating asymmetric power updates

Input: Tensor $T \in \mathbb{R}^{d\times d\times d}$, number of initializations L, number of iterations N.

1: **for** $\tau = 1$ **to** L **do**

2: **Initialize** unit vectors $\widehat{a}_\tau^{(0)} \in \mathbb{R}^d$, $\widehat{b}_\tau^{(0)} \in \mathbb{R}^d$, and $\widehat{c}_\tau^{(0)} \in \mathbb{R}^d$ as

- Option 1: SVD-based method in Procedure 2 when $k \le \beta d$ for arbitrary constant β.
- Option 2: random initialization.

3: **for** $t = 0$ **to** $N-1$ **do**

4: Asymmetric power updates (see (1.3) for the definition of the multilinear form):

$$\widehat{a}_\tau^{(t+1)} = \frac{T\left(I, \widehat{b}_\tau^{(t)}, \widehat{c}_\tau^{(t)}\right)}{\left\|T\left(I, \widehat{b}_\tau^{(t)}, \widehat{c}_\tau^{(t)}\right)\right\|}, \quad \widehat{b}_\tau^{(t+1)} = \frac{T\left(\widehat{a}_\tau^{(t)}, I, \widehat{c}_\tau^{(t)}\right)}{\left\|T\left(\widehat{a}_\tau^{(t)}, I, \widehat{c}_\tau^{(t)}\right)\right\|}, \quad \widehat{c}_\tau^{(t+1)} = \frac{T\left(\widehat{a}_\tau^{(t)}, \widehat{b}_\tau^{(t)}, I\right)}{\left\|T\left(\widehat{a}_\tau^{(t)}, \widehat{b}_\tau^{(t)}, I\right)\right\|}.$$

5: weight estimation:

$$\widehat{w}_\tau = T\left(\widehat{a}_\tau^{(N)}, \widehat{b}_\tau^{(N)}, \widehat{c}_\tau^{(N)}\right). \tag{2.8}$$

6: Cluster set $\left\{\left(\widehat{w}_\tau, \widehat{a}_\tau^{(N)}, \widehat{b}_\tau^{(N)}, \widehat{c}_\tau^{(N)}\right), \tau \in [L]\right\}$ into k clusters as in Procedure 3.

7: **return** the center member of these k clusters as estimates $(\widehat{w}_j, \widehat{a}_j, \widehat{b}_j, \widehat{c}_j), j \in [k]$.

where $\mathcal{S}^{d-1}$ denotes the unit d-dimensional sphere. This optimization program is non-convex, and has multiple local optima. It can be shown that the updates in (2.6) are the alternating optimization for this program where in each update, optimization over one vector is performed while the other two vectors are assumed fixed. This alternating minimization approach does not converge to the true components of tensor T in general, and in this work we provide sufficient conditions for the convergence guarantees.

Intuition: We now provide an intuitive argument on the functionality of power updates in (2.6). Consider a rank-k tensor T as in (1.5), and suppose we start at the correct vectors $\widehat{a} = a_j$ and $\widehat{b} = b_j$, for some $j \in [k]$. Then for the numerator of update formula (2.6), we have

$$T\left(\widehat{a}, \widehat{b}, I\right) = T\left(a_j, b_j, I\right) = w_j c_j + \sum_{i\neq j} w_i \langle a_j, a_i\rangle \langle b_j, b_i\rangle c_i, \tag{2.9}$$

where the first term is along c_j and the second term is an error term due to non-orthogonality. For orthogonal decomposition, the second term is zero, and the true vectors a_j, b_j and c_j are stationary points for the power update procedure. However, since we consider non-orthogonal tensors, this

Procedure 2 SVD-based initialization when $k \leq \beta d$ for arbitrary constant β

Input: Tensor $T \in \mathbb{R}^{d \times d \times d}$.

1: Draw a random standard Gaussian vector $\theta \sim \mathcal{N}(0, I_d)$.
2: Compute u_1 and v_1 as the top left and right singular vectors of $T(I, I, \theta) \in \mathbb{R}^{d \times d}$.
3: $\widehat{a}^{(0)} \leftarrow u_1$, $\widehat{b}^{(0)} \leftarrow v_1$.
4: Initialize $\widehat{c}^{(0)}$ by update formula in (2.6).
5: **return** $\left(\widehat{a}^{(0)}, \widehat{b}^{(0)}, \widehat{c}^{(0)}\right)$.

procedure cannot recover the decomposition exactly leading to a residual error after running this step. Under incoherence conditions which encourages soft-orthogonality constraints [4] (and some other conditions), we show that the residual error is small (see Lemma 2.2 where the guarantees for the tensor power iteration step is provided), and thus, with the additional step we propose in Section 2.3.2, we can also remove this residual error.

Initialization and clustering procedures: We discussed that the tensor power updates in (2.6) are the alternating iterations for the problem of rank-1 approximation of the tensor; see (2.7). This is a non-convex problem and has many local optima. Thus, the power update requires careful initialization to ensure convergence to the true rank-1 tensor components.

For generating initialization vectors $\left(\widehat{a}^{(0)}, \widehat{b}^{(0)}, \widehat{c}^{(0)}\right)$, we introduce two possibilities. One is the simple random initializations, where $\widehat{a}^{(0)}$ and $\widehat{b}^{(0)}$ are uniformly drawn from unit sphere $\mathcal{S}^{d-1}$. The other option is SVD-based technique in Procedure 2 where top left and right singular vectors of $T(I, I, \theta)$ (for some random $\theta \in \mathbb{R}^d$) are respectively introduced as $\widehat{a}^{(0)}$ and $\widehat{b}^{(0)}$. Under both initialization procedures, vector $\widehat{c}^{(0)}$ is generated through update formula in (2.6). We establish in Section 2.4.2 that when $k = O(d)$, the SVD procedure leads to global convergence guarantees under polynomial number of trials. In practice random initialization also works well, however the analysis is still an open problem.

Notice that the algorithm is run for L different initialization vectors for which we do not know the good ones in prior. In order to identify which initializations are successful at the end, we also need a *clustering* step proposed in Procedure 3 to obtain the final estimates of the vectors. The detailed analysis of clustering procedure is provided in Appendix A.5.

[4] See Assumption (A2) in Appendix A.2 for precise description.

Procedure 3 Clustering process

Input: Tensor $T \in \mathbb{R}^{d\times d\times d}$, set of 4-tuples $\Big\{(\widehat{w}_\tau, \widehat{a}_\tau, \widehat{b}_\tau, \widehat{c}_\tau), \tau \in [L]\Big\}$, parameter ν.

1: **for** $i = 1$ **to** k **do**
2: Among the remaining 4-tuples, choose $\widehat{a}, \widehat{b}, \widehat{c}$ which correspond to the largest $|T(\widehat{a}, \widehat{b}, \widehat{c})|$.
3: Do N more iterations of alternating updates in (2.6) starting from $\widehat{a}, \widehat{b}, \widehat{c}$.
4: Let the output of iterations denoted by $(\widehat{a}, \widehat{b}, \widehat{c})$ be the center of cluster i.
5: Remove all the tuples with $\max\{|\langle \widehat{a}_\tau, \widehat{a}\rangle|, |\langle \widehat{b}_\tau, \widehat{b}\rangle|, |\langle \widehat{c}_\tau, \widehat{c}\rangle|\} > \nu/2$.
6: **return** the k cluster centers.

2.3.2 Coordinate descent iteration in Algorithm 4

We discussed in the previous section that the tensor power iteration recovers the tensor rank-1 components up to some residual error. We now propose Algorithm 4 to remove this additional residual error. This algorithm mainly runs a coordinate descent iteration as

$$\tilde{c}_i^{(t+1)} = \mathrm{Norm}\bigg(T\left(\widehat{a}_i^{(t)}, \widehat{b}_i^{(t)}, I\right) - \sum_{j\neq i} \widehat{w}_j^{(t)} \langle \widehat{a}_i^{(t)}, \widehat{a}_j^{(t)}\rangle \langle \widehat{b}_i^{(t)}, \widehat{b}_j^{(t)}\rangle \cdot \widehat{c}_j^{(t)}\bigg), \quad i \in [k], \tag{2.10}$$

where for vector v, we have $\mathrm{Norm}(v) := v/\|v\|$, i.e., it normalizes the vector. The above is similarly applied for updating $\tilde{a}_i^{(t+1)}$ and $\tilde{b}_i^{(t+1)}$. Unlike the power iteration, it can be immediately seen that a_i, b_i and c_i are stationary points of the above update even if the components are not orthogonal to each other. Inspired by this intuition, we prove that when the residual error is small enough (as guaranteed in the analysis of tensor power iteration), this step removes it.

The analysis of this algorithm requires that the estimate matrices $\widehat{A}, \widehat{B}, \widehat{C}$ satisfy some bound on the spectral norm and some column-wise error bounds; see Definition A.1 in Appendix A.3.2 for the details. The optimization program in (2.11) (which is only run in the first iteration) and projection Procedure 5 ensure that these conditions are satisfied.

2.3.3 Discussions

We now provide some further discussions and comparisons about the algorithm.

Algorithm 4 Coordinate descent algorithm for removing the residual error

Input: Tensor $T \in \mathbb{R}^{d\times d\times d}$, initialization set $\left\{\widehat{A}, \widehat{B}, \widehat{C}, \widehat{w}^{(0)}\right\}$, number of iterations N.

1: Initialize $\widehat{A}^{(0)}$ as (similarly for $\widehat{B}^{(0)}, \widehat{C}^{(0)}$)

$$\widehat{A}^{(0)} := \arg\min_{\tilde{A}} \|\tilde{A}\| \quad \text{s. t. } \|\tilde{a}_i - \widehat{a}_i\| \leq \tilde{O}\left(\sqrt{k}/d\right), i \in [k]. \tag{2.11}$$

2: **for** $t = 0$ **to** $N-1$ **do**

3: **for** $i = 1$ **to** k **do**

4:

$$\tilde{w}_i^{(t+1)} = \left\| T\left(\widehat{a}_i^{(t)}, \widehat{b}_i^{(t)}, I\right) - \sum_{j\neq i} \widehat{w}_j^{(t)} \langle \widehat{a}_i^{(t)}, \widehat{a}_j^{(t)}\rangle \langle \widehat{b}_i^{(t)}, \widehat{b}_j^{(t)}\rangle \cdot \widehat{c}_j^{(t)} \right\|,$$

$$\tilde{c}_i^{(t+1)} = \frac{1}{\tilde{w}_i^{(t+1)}} \left(T\left(\widehat{a}_i^{(t)}, \widehat{b}_i^{(t)}, I\right) - \sum_{j\neq i} \widehat{w}_j^{(t)} \langle \widehat{a}_i^{(t)}, \widehat{a}_j^{(t)}\rangle \langle \widehat{b}_i^{(t)}, \widehat{b}_j^{(t)}\rangle \cdot \widehat{c}_j^{(t)} \right).$$

5: Update $\widehat{C}^{(t+1)}$ by applying Procedure 5 with inputs $\tilde{C}^{(t+1)}$ and $\widehat{C}^{(t)}$.

6: Repeat the above steps (with appropriate changes) to update $\widehat{A}^{(t+1)}$ and $\widehat{B}^{(t+1)}$.

7: Update $\widehat{w}^{(t+1)}$:

$$\text{for any } i \in [k],\ \widehat{w}_i^{(t+1)} = \begin{cases} \tilde{w}_i^{(t+1)}, & \left|\tilde{w}_i^{(t+1)} - \widehat{w}_i^{(t)}\right| \leq \eta_0 \frac{\sqrt{k}}{d}, \\ \widehat{w}_i^{(t)} + \operatorname{sgn}\left(\tilde{w}_i^{(t+1)} - \widehat{w}_i^{(t)}\right) \cdot \eta_0 \frac{\sqrt{k}}{d}, & \text{o. w.} \end{cases}$$

8: **return** $\left\{\widehat{A}^{(N)}, \widehat{B}^{(N)}, \widehat{C}^{(N)}, \widehat{w}^{(N)}\right\}$.

Implicit tensor operations: In many applications, the input tensor T is not available in advance, and it is computed from samples. It is discussed in [21] that the tensor is not needed to be computed and stored explicitly, where the multilinear tensor updates (2.6) and (2.10) in the algorithm can be efficiently computed through multilinear operations on the samples directly.

Comparison with symmetric orthogonal tensor power method: Algorithm 1 is similar to the symmetric tensor power method analyzed by Anandkumar et al. [15] with the following main differences, viz.,

- Symmetric and non-symmetric tensors: Our algorithm can be applied to both symmetric and non-symmetric tensors, while tensor power method in Anandkumar et al. [15] is only for symmetric tensors.
- Linearity: The updates in Algorithm 1 are linear in each variable, while the symmetric tensor power update is a quadratic operator given a third order tensor.

Procedure 5 Projection procedure

input Matrices $\tilde{C}^{(t+1)}$, $\widehat{C}^{(t)}$.

1: Compute the SVD of $\tilde{C}^{(t+1)} = UDV^\top$.

2: Let $\widehat{D}$ be the truncated version of D as $\widehat{D}_{i,i} := \min\left\{D_{i,i}, \eta_1\sqrt{\frac{k}{d}}\right\}$.

3: Let $Q := U\widehat{D}V^\top$.

4: Update $\widehat{C}^{(t+1)}$: for any $i \in [k]$, $\widehat{c}_i^{(t+1)} = \begin{cases} Q_i, & \left\|Q_i - \widehat{c}_i^{(t)}\right\| \leq \eta_0\frac{\sqrt{k}}{d}, \\ \widehat{c}_i^{(t)} + \eta_0\frac{\sqrt{k}}{d}\frac{\left(Q_i - \widehat{c}_i^{(t)}\right)}{\left\|Q_i - \widehat{c}_i^{(t)}\right\|}, & \text{o. w.} \end{cases}$

5: **return** $\widehat{C}^{(t+1)}$.

- Guarantees: In Anandkumar et al. [15], guarantees for the symmetric tensor power update under orthogonality are obtained, while here we consider non-orthogonal tensors under the alternating updates.

Comparison with Alternating Least Square(ALS): The updates in Algorithm 1 can be viewed as a rank-1 form of the standard alternating least squares (ALS) procedure. This is because the unnormalized update for c in (2.6) can be rewritten as

$$\tilde{c}_\tau^{(t+1)} := T\left(\widehat{a}_\tau^{(t)}, \widehat{b}_\tau^{(t)}, I\right) = \mathrm{mat}(T,3)\cdot\left(\widehat{b}_\tau^{(t)} \odot \widehat{a}_\tau^{(t)}\right), \tag{2.12}$$

where $\odot$ denotes the *Khatri-Rao* product, and $\mathrm{mat}(T,3) \in \mathbb{R}^{d\times d^2}$ is the mode-3 matricization of tensor T. On the other hand, the ALS update has the form

$$\tilde{C}^{(t+1)} = \mathrm{mat}(T,3)\cdot\left(\left(\widehat{B}^{(t)} \odot \widehat{A}^{(t)}\right)^\top\right)^\dagger,$$

where k vectors (all columns of $\tilde{C}^{(t+1)} \in \mathbb{R}^{d\times k}$) are simultaneously updated given the current estimates for the other two modes $\widehat{A}^{(t)}$ and $\widehat{B}^{(t)}$. In contrast, our procedure updates only one vector (with the target of recovering one column of C) in each iteration. In our update, we do not require finding matrix inverses. This leads to efficient computational complexity, and we also show that our update procedure is more robust to perturbations.

2.4 Guarantees for Tensor Decomposition Under Incoherent Components

In this section, we provide the local and global convergence guarantees for the tensor decomposition algorithm proposed in Section 2.3. A summary of these results is proposed in Section 2.1. Throughout the work, we assume tensor $\widehat{T} \in \mathbb{R}^{d\times d\times d}$ is of the form $\widehat{T} = T + \Psi$, where Ψ is the error or perturbation tensor, and[5]

$$T = \sum_{i\in[k]} w_i \cdot a_i \otimes b_i \otimes c_i,$$

is a rank-k tensor such that $a_i, b_i, c_i \in \mathbb{R}^d, i \in [k]$, are unit vectors. Let $A := [a_1\ a_2\ \cdots\ a_k] \in \mathbb{R}^{d\times k}$, and B and C are similarly defined. The goal of robust tensor decomposition algorithm is to recover the rank-1 components $\{(a_i, b_i, c_i), i \in [k]\}$ given noisy tensor $\widehat{T}$. Our analysis emphasizes on the challenging *overcomplete* regime where the tensor rank is larger than the dimension, i.e., $k > d$. Without loss of generality we also assume $w_{\max} = w_1 \geq w_2 \geq \cdots \geq w_k = w_{\min} > 0$.

We require natural deterministic conditions on the tensor components to argue the convergence guarantees; see Appendix A.2 for the details. We show that all of these conditions are satisfied if the true rank-1 components of the tensor are uniformly i.i.d. drawn from the unit d-dimensional sphere $\mathcal{S}^{d-1}$. Thus, for simplicity we assume this random assumption in the main part, and state the deterministic assumptions in Appendix A.2. Notice that it is also reasonable to assume these deterministic assumptions hold for some non-random matrices. Among the deterministic assumptions, the most important one is the *incoherence* condition which imposes a soft-orthogonality constraint between different rank-1 components of the tensor.

Some of the convergence guarantees are provided in terms of distance between the estimated and the true vectors, defined below.

[5]For 4th and higher order tensors, same techniques we introduce in this work, can be exploited to argue similar results.

Definition 2.1. *For any two vectors $u, v \in \mathbb{R}^d$, the distance between them is defined as*

$$\text{dist}(u, v) := \sup_{z \perp u} \frac{\langle z, v \rangle}{\|z\| \cdot \|v\|} = \sup_{z \perp v} \frac{\langle z, u \rangle}{\|z\| \cdot \|u\|}. \tag{2.13}$$

Note that distance function $\text{dist}(u, v)$ is invariant w.r.t. norm of input vectors u and v. Distance also provides an upper bound on the error between unit vectors u and v as (see Lemma A.1 of Agarwal et al. [3])

$$\min_{z \in \{-1, 1\}} \|zu - v\| \leq \sqrt{2}\, \text{dist}(u, v).$$

Incorporating distance notion resolves the sign ambiguity issue in recovering the components: note that a third order tensor is unchanged if the sign of a vector along one of the modes is fixed and the signs of the corresponding vectors in the other two modes are flipped.

2.4.1 Local convergence guarantee

In the local convergence guarantee, we analyze the convergence properties of the algorithm assuming we have good initialization vectors for the non-convex tensor decomposition algorithm.

Settings of Algorithm in Theorem 2.4:

- Number of iterations: $N = \Theta\left(\log\left(\frac{1}{\gamma \epsilon_R}\right)\right)$, where $\gamma := \frac{w_{\max}}{w_{\min}}$ and $\epsilon_R := \min\left\{\frac{\psi}{w_{\min}}, \tilde{O}\left(\gamma \frac{\sqrt{k}}{d}\right)\right\}$.

Conditions for Theorem 2.4:

- Rank-k true tensor with random components: Let

$$T = \sum_{i \in [k]} w_i \cdot a_i \otimes b_i \otimes c_i, \quad w_i > 0, a_i, b_i, c_i \in \mathcal{S}^{d-1},$$

where $a_i, b_i, c_i, i \in [k]$, are uniformly i.i.d. drawn from the unit d-dimensional sphere $\mathcal{S}^{d-1}$. We state the deterministic assumptions in Appendix A.2, and show that random matrices satisfy these assumptions.

- Rank condition: $k = o\left(d^{1.5}\right)$.
- Perturbation tensor Ψ satisfies the bound

$$\psi := \|\Psi\| \leq \frac{w_{\min}}{6}.$$

- Weight ratio: The maximum ratio of weights $\gamma := \frac{w_{\max}}{w_{\min}}$ satisfies the bound

$$\gamma = O\left(\min\left\{\sqrt{d}, \frac{d^{1.5}}{k}\right\}\right).$$

- Initialization: Assume we have good initialization vectors $\widehat{a}_j^{(0)}, \widehat{b}_j^{(0)}, j \in [k]$ satisfying

$$\epsilon_0 := \max\left\{\mathrm{dist}\left(\widehat{a}_j^{(0)}, a_j\right), \mathrm{dist}\left(\widehat{b}_j^{(0)}, b_j\right)\right\} = O(1/\gamma), \quad \forall j \in [k], \tag{2.14}$$

where $\gamma := \frac{w_{\max}}{w_{\min}}$. In addition, given $\widehat{a}_j^{(0)}$ and $\widehat{b}_j^{(0)}$, suppose $\widehat{c}_j^{(0)}$ is also calculated by the update formula in (2.6).

Theorem 2.4 (Local convergence guarantee of the tensor decomposition algorithm)**.** *Consider noisy rank-k tensor $\widehat{T} = T + \Psi$ as the input to the tensor decomposition algorithm, and assume the conditions and settings mentioned above hold. Then the algorithm outputs estimates $\widehat{A} := [\widehat{a}_1 \cdots \widehat{a}_k] \in \mathbb{R}^{d\times k}$ and $\widehat{w} := [\widehat{w}_1 \cdots \widehat{w}_k]^\top \in \mathbb{R}^k$, satisfying w.h.p.*

$$\left\|\widehat{A} - A\right\|_F \leq \tilde{O}\left(\frac{\sqrt{k}\cdot\psi}{w_{\min}}\right), \quad \|\widehat{w} - w\| \leq \tilde{O}\left(\sqrt{k}\cdot\psi\right).$$

Same error bounds hold for other factor matrices $B := [b_1 \cdots b_k]$ and $C := [c_1 \cdots c_k]$.

See the proof in Appendix A.3.

Thus, we can efficiently decompose the tensor in the highly overcomplete regime $k \leq o\left(d^{1.5}\right)$ under incoherent factors and some other assumptions mentioned above. The deterministic version of

assumptions are stated in Appendix A.2. We show that these assumptions are true for random components which is assumed here for simplicity. If k is significantly smaller than $d^{1.5}$ ($k \ll d^{1.25}$), then many of the assumptions can be derived from incoherence. See Appendix A.2 for the details.

The above local convergence result can be also interpreted as a local identifiability result for tensor decomposition under incoherent factors.

The $\sqrt{k}$ factor in the above theorem error bound is from the fact that the final recovery guarantee is on the Frobenius norm of the whole factor matrix A. In the following, we provide stronger column-wise guarantees (where there is no $\sqrt{k}$ factor) with the expense of having an additional residual error term. Recall that our algorithm includes two main update steps including tensor power iteration in (2.6) and residual error removal in (2.10). The guarantee for the first step — tensor power iteration — is provided in the following lemma.

Lemma 2.2 (Local convergence guarantee of the tensor power updates, Algorithm 1)**.** *Consider the same settings as in Theorem 2.4. Then, the outputs of tensor power iteration steps (output of Algorithm 1) satisfy w.h.p.*

$$\mathrm{dist}(\widehat{a}_j, a_j) \leq \tilde{O}\left(\frac{\psi}{w_{\min}}\right) + \tilde{O}\left(\gamma\frac{\sqrt{k}}{d}\right), \quad |\widehat{w}_j - w_j| \leq \tilde{O}\left(\psi\right) + \tilde{O}\left(w_{\max}\frac{\sqrt{k}}{d}\right), \quad j \in [k].$$

Same error bounds hold for other factor matrices B and C.

The above result provides guarantees with the additional residual error $\tilde{O}\left(\gamma\frac{\sqrt{k}}{d}\right)$, but we believe this result also has independent importance for the following reasons. The above result provides column-wise guarantees which is stronger than the guarantees on the whole factor matrix in Theorem 2.4. Furthermore, we can only have recovery guarantees for a subset of rank-1 components of the tensor (the ones for which we have good initializations) without worrying about the rest of components. Finally, in the high-dimensional regime (large d), the residual error term goes to zero.

The result in the above lemma is actually stated in the non-asymptotic form, where the details of constants are explicitly provided in Appendix A.2.

Symmetric tensor decomposition: The above local convergence result also holds for recovering the components of a rank-k *symmetric* tensor. Consider symmetric tensor T with CP decomposition $T = \sum_{i\in[k]} w_i a_i \otimes a_i \otimes a_i$. The proposed algorithm can be also applied to recover the components $a_i, i \in [k]$, where the main updates are changed to adapt to the symmetric tensor. The tensor power iteration is changed to

$$\widehat{a}^{(t+1)} = \frac{T\left(\widehat{a}^{(t)}, \widehat{a}^{(t)}, I\right)}{\left\|T\left(\widehat{a}^{(t)}, \widehat{a}^{(t)}, I\right)\right\|}, \tag{2.15}$$

and the coordinate descent update is changed to the form stated in (A.10). Then, the same local convergence result as in Theorem 2.4 holds for this algorithm. The proof is very similar to the proof of Theorem 2.4 with some slight modifications considering the symmetric structure.

Extension to higher order tensors: We also provide the generalization of the tensor decomposition guarantees to higher order tensors. We state and prove the result for the tensor power iteration part in details, while the generalization of coordinate descent part (for removing the residual error) to higher order tensors, can be argued by the same techniques we introduce in this work

For brevity, Algorithm 1 and local convergence guarantee in Lemma 2.2 are provided for a 3rd order tensor. The algorithm can be simply extended to higher order tensors to compute the corresponding CP decomposition. Consider p-th order tensor $T \in \bigotimes^p \mathbb{R}^d$ with CP decomposition

$$T = \sum_{i\in[k]} w_i \cdot a_{(1),i} \otimes a_{(2),i} \otimes \cdots \otimes a_{(p),i}, \tag{2.16}$$

where $a_{(r),i} \in \mathbb{R}^d$ is the i-th column of r-th component $A_{(r)} := \left[a_{(r),1}\ a_{(r),2}\ \cdots\ a_{(r),k}\right] \in \mathbb{R}^{d\times k}$, for $r \in [p]$. Algorithm 1 can be extended to recover the components of above decomposition where update formula for the p-th mode is modified as

$$\widehat{a}_{(p)}^{(t+1)} = \frac{T\left(\widehat{a}_{(1)}^{(t)}, \widehat{a}_{(2)}^{(t)}, \ldots, \widehat{a}_{(p-1)}^{(t)}, I\right)}{\left\|T\left(\widehat{a}_{(1)}^{(t)}, \widehat{a}_{(2)}^{(t)}, \ldots, \widehat{a}_{(p-1)}^{(t)}, I\right)\right\|}, \tag{2.17}$$

and similarly the other updates are changed. Then, we have the following generalization of Lemma 2.2 to higher order tensors.

Corollary 2.1 (Local convergence guarantee of the tensor power updates in Algorithm 1 for p-th order tensor)**.** *Consider the same conditions and settings as in Lemma 2.2, unless tensor T is p-th order with CP decomposition in* (2.16) *where $p \geq 3$ is a constant. In addition, the bounds on $\gamma := \frac{w_{\max}}{w_{\min}}$ and k are modified as*

$$\gamma = O\left(\min\left\{d^{\frac{p-2}{2}}, \frac{d^{p/2}}{k}\right\}\right), \quad k = o\left(d^{\frac{p}{2}}\right).$$

Then, the outputs of tensor power iteration steps (output of Algorithm 1) satisfy w.h.p.

$$\text{dist}\left(\widehat{a}_{(r),j}, a_{(r),j}\right) \leq \tilde{O}\left(\frac{\psi}{w_{\min}}\right) + \tilde{O}\left(\gamma\sqrt{\frac{k}{d^{p-1}}}\right), \quad |\widehat{w}_j - w_j| \leq \tilde{O}\left(\psi\right) + \tilde{O}\left(w_{\max}\sqrt{\frac{k}{d^{p-1}}}\right),$$

for $j \in [k]$ and $r \in [p]$. The number of iterations is $N = \Theta\left(\log\left(\frac{1}{\gamma\tilde{\epsilon}_R}\right)\right)$, where $\gamma := \frac{w_{\max}}{w_{\min}}$ and $\tilde{\epsilon}_R := \min\left\{\frac{\psi}{w_{\min}}, \tilde{O}\left(\gamma\sqrt{k/d^{p-1}}\right)\right\}$.

2.4.2 Global convergence guarantee when $k = O(d)$

Theorem 2.4 provides local convergence guarantee given good initialization vectors. In this section, we exploit SVD-based initialization method in Procedure 2 to provide good initialization vectors when $k = O(d)$. This method proposes the top singular vectors of random slices of the moment tensor as the initialization. Combining the theoretical guarantees of this initialization method (provided in Appendix A.4) with the local convergence guarantee in Theorem 2.4, we provide the following global convergence result.

Settings of Algorithm in Theorem 2.5:

- Number of iterations: $N = \Theta\left(\log\left(\frac{1}{\gamma\epsilon_R}\right)\right)$, where $\gamma := \frac{w_{\max}}{w_{\min}}$ and $\epsilon_R := \min\left\{\frac{\psi}{w_{\min}}, \tilde{O}\left(\gamma\frac{\sqrt{k}}{d}\right)\right\}$.
- The initialization in each run of Algorithm 1 is performed by SVD-based technique proposed in Procedure 2, with the number of initializations as

$$L \geq k^{\Omega\left(\gamma^4(k/d)^2\right)}.$$

Conditions for Theorem 2.5:

- Rank-k decomposition and perturbation conditions as [6]

$$T = \sum_{i \in [k]} w_i \cdot a_i \otimes b_i \otimes c_i, \quad \psi := \|\Psi\| \leq \frac{w_{\min}\sqrt{\log k}}{\alpha_0 \sqrt{d}},$$

  where $a_i, b_i, c_i, i \in [k]$, are uniformly i.i.d. drawn from the unit d-dimensional sphere $\mathcal{S}^{d-1}$, and $\alpha_0 > 1$ is a constant.

- Rank condition: $k = O(d)$, i.e., $k \leq \beta d$ for arbitrary constant $\beta > 1$.

Theorem 2.5 (Global convergence guarantee of tensor decomposition algorithm when $k = O(d)$)**.** *Consider noisy rank-k tensor $\widehat{T} = T + \Psi$ as the input to the tensor decomposition algorithm, and assume the conditions and settings mentioned above hold. Then, the same guarantees as in Theorem 2.4 hold.*

See the proof in Appendix A.3.

Thus, we can efficiently recover the tensor decomposition, when the tensor is undercomplete or mildly overcomplete (i.e., $k \leq \beta d$ for arbitrary constant $\beta > 1$), by initializing the algorithm with a simple SVD-based technique. The number of initialization trials L is polynomial when γ is a constant, and $k = O(d)$.

Note that the argument in Lemma 2.2 can be similarly adapted leading to global convergence guarantee of the tensor power iteration step.

Two undercomplete, and one overcomplete component

Here, we apply the global convergence result to the regime of two undercomplete and one overcomplete components. This arises in supervised learning problems under a multiview mixtures model and employing moment tensor $\mathbb{E}[x_1 \otimes x_2 \otimes y]$, where $x_i \in \mathbb{R}^{d_u}$ are multi-view high-dimensional features and $y \in \mathbb{R}^{d_o}$ is a low-dimensional label.

[6] Note that the perturbation condition is stricter than the corresponding condition in the local convergence guarantee (Theorem 2.4).

Since in the SVD initialization Procedure 2, two components $\widehat{a}^{(0)}$ and $\widehat{b}^{(0)}$ are initialized through SVD, and the third component $\widehat{c}^{(0)}$ is initialized through update formula (2.6), we can generalize the global convergence result in Theorem 2.5 to the setting where A, B are undercomplete, and C is overcomplete.

Corollary 2.2. *Consider the same setting as in Theorem 2.5. In addition, suppose the regime of undercomplete components $A \in \mathbb{R}^{d_u \times k}$, $B \in \mathbb{R}^{d_u \times k}$, and overcomplete component $C \in \mathbb{R}^{d_o \times k}$ such that $d_u \geq k \geq d_o$. In addition, in this case the bound on $\gamma := \frac{w_{\max}}{w_{\min}}$ is*

$$\gamma = O\left(\min\left\{\sqrt{d_o}, \frac{d_u\sqrt{d_o}}{k}\right\}\right).$$

Then, if $k = O(d_u)$ and $d_o \geq \mathrm{polylog}(k)$, the same convergence guarantee as in Theorem 2.5 holds.

See the proof in Appendix A.3.

We observe that given undercomplete modes A and B, mode C can be arbitrarily overcomplete, and we can still provide global recovery of A, B and C by employing SVD initialization procedure along modes A and B.

2.5 Proof Outline Under Incoherent Components

The global convergence guarantee in Theorem 2.5 is established by combining the local convergence result in Theorem 2.4 and the SVD initialization result in Appendix A.4.

The local convergence result in Theorem 2.4 is derived by establishing error contraction in each iteration of the tensor power iteration and the coordinate descent for removing the residual error. Note that these convergence properties are broken down in Lemmata 2.2 and A.11, respectively.

Since we assume generic factor matrices A, B and C, we utilize many useful properties such as incoherence, bounded spectral norm of the matrices A, B and C, bounded tensor spectral norm and so on. We list the precise set of deterministic conditions required to establish the local convergence result in Appendix A.2. Under these conditions, with a good initialization (i.e., small enough

$\max\{\text{dist}(\widehat{a}, a_j), \text{dist}(\widehat{b}, b_j)\} \leq \epsilon_0)$, we show that the iterative update in (2.6) provides an estimate $\widehat{c}$ with

$$\text{dist}(\widehat{c}, c_j) < \tilde{O}\left(\frac{\psi}{w_{\min}}\right) + \tilde{O}\left(\gamma\frac{\sqrt{k}}{d}\right) + q\epsilon_0,$$

for some contraction factor $q < 1/2$. The incoherence condition is crucial for establishing this result. See Appendix A.3 for the complete proof.

The initialization argument for SVD-based technique in Procedure 2 has two parts. The first part claims that by performing enough number of initializations (large enough L), a gap condition is satisfied, meaning that we obtain a vector θ which is relatively close to c_j compared to any $c_i, i \neq j$. This is a standard result for Gaussian vectors, e.g., see Lemma B.1 of Anandkumar et al. [15]. In the second part of the argument, we analyze the dominant singular vectors of $T(I, I, \theta)$, for a vector θ with a good relative gap, to obtain an error bound on the initialization vectors. This is obtained through standard matrix perturbation results (Weyl and Wedin's theorems). See Appendix A.4 for the complete proof.

2.6 Proof Outline Under Random Components

Our main technical result is the analysis of third order tensor power iteration provided in Theorem 2.3 which also allows to tolerate some amount of noise in the input tensor. We analyze the noiseless and noisy settings in different ways. We basically first prove the result for the noiseless setting where the input tensor has an exact rank-k decomposition in (2.1). When the noise is also considered, we show that the contribution of noise in the analysis is dominated by the main signal, and thus, the same result still holds. For the rest of this section we focus on the noiseless setting, while we discuss the proof ideas for the noisy case in Section 2.6.2.

We first discuss the proof of Theorem 3.10 which involves two phases. In the first phase, we show that under certain small amount of correlation (see (2.19)) between the initial vector and the true component, the power iteration in (2.2) converges to some vector which has constant correlation

with the true component. This result is the core technical analysis of this work which is provided in Lemma 2.3. In the second phase, we incorporate the result of Anandkumar et al. [19] which guarantees the approximate convergence of power iteration given initial vector having constant correlation with the true component. This is stated in Lemma 2.4.

To simplify the notation, we consider the tensor [7]

$$T = \sum_{j \in [k]} a_j \otimes a_j \otimes a_j, \quad a_j \sim \mathcal{N}(0, \frac{1}{d} I_d). \tag{2.18}$$

Notice that this is exactly proportional to the 3rd order moment tensor of the multiview mixture model in (3.2).

The following lemma is restatement of Theorem 2.3 in the noiseless setting.

Lemma 2.3 (Dynamics of tensor power iteration, phase 1)**.** *Consider the rank-k tensor T of the form in* (2.18)*. Let tensor rank $k = o(d^{1.5})$, and the unit-norm initial vector $x^{(1)}$ satisfies the correlation bound*

$$|\langle x^{(1)}, a_j \rangle| \geq d^{\beta} \frac{\sqrt{k}}{d}, \tag{2.19}$$

w.r.t. some true component $a_j, j \in [k]$, for some $\beta > (\log d)^{-c}$ for some universal constant $c > 0$. After $N = \Theta(\log \log d)$ iterations, the tensor power iteration in (2.2) *outputs a vector having w.h.p. a constant correlation with the true component a_j as*

$$|\langle x^{(N+1)}, a_j \rangle| \geq 1 - \gamma,$$

for any fixed constant $\gamma > 0$.

The proof outline of above lemma is provided in Section 2.6.1.

Lemma 2.4 (Dynamics of tensor power iteration, phase 2 [19])**.** *Consider the rank-k tensor T of the form in* (2.18) *with rank condition $k \leq o(d^{1.5})$. Let the initial vectors $x_j^{(1)}$ satisfy the constant*

[7] In the analysis, we assume that all the weights are equal to one which can be generalized to the case when the ratio of maximum and minimum weights (in absolute value) are constant.

correlation bound

$$|\langle x_j^{(1)}, a_j\rangle| \geq 1 - \gamma_j,$$

w.r.t. true components $a_j, j \in [k]$*, for some constants* $\gamma_j > 0$*. Let the output of the tensor power update*[8] *in* (2.2) *applied to all these different initialization vectors after* $N = \Theta\left(\log \frac{1}{\epsilon}\right)$ *iterations be stacked in matrix* $\widehat{A}$*. Then, we have w.h.p.*[9]

$$\left\|\widehat{A} - A\right\|_F \leq \epsilon.$$

Given the above two lemmas, the learning result in Theorem 3.10 is directly proved.

Proof of Theorem 3.10: The result is proved by combining Lemma 2.3 and Lemma 2.4. Note that the initialization condition in (2.5) is w.h.p. satisfied given the SNR bound assumed. □

2.6.1 Proof outline of Lemma 2.3 (noiseless case of Theorem 2.3)

First step: We first intuitively show the first step of the algorithm makes progress. Suppose the tensor is $T = \sum_{j\in[k]} a_j \otimes a_j \otimes a_j$, and the initial vector x has correlation $|\langle x, a_1\rangle| \geq d^{\beta}\frac{\sqrt{k}}{d}$ with the first component. The result of the first iteration is the normalized version of the following vector:

$$\tilde{x} = \sum_{j\in[k]} \langle a_j, x\rangle^2 a_j.$$

Intuitively, this vector should have roughly $\langle a_1, \tilde{x}\rangle = d^{2\beta}\frac{k}{d^2}$ correlation with a_1 (as the other terms are random they don't contribute much). On the other hand, the norm of this vector is roughly $O(\sqrt{k}/d)$: this is because $\langle a_j, x\rangle^2$ for $j \neq 1$ is roughly[10] $1/d$, and the sum of k random vectors with

[8] This result also needs an additional step of coordinate descent iterations since the true components are not the fixed points of power iteration; see Anandkumar et al. [19] for the details.

[9] Anandkumar et al. [19] recover the vector up to sign since they work in the asymmetric case. In symmetric case it is easy to resolve sign ambiguity issue.

[10] The correlation between two unit Gaussian vectors in d dimensions is roughly $1/\sqrt{d}$.

length $1/d$ will have length roughly $O(\sqrt{k}/d)$. These arguments can be made precise showing the normalized version $\tilde{x}/\|\tilde{x}\|$ has correlation $d^{2\beta}\frac{\sqrt{k}}{d}$ with a_1 ensuring progress in the first step.

Going forward: As we explained, the basic idea behind proving Lemma 2.3 is to characterize the conditional distribution of random Gaussian tensor components a_j's given previous iterations. In particular, we show that the residual independent randomness left in these conditional distributions is large enough and we can exploit it to obtain tighter concentration bounds throughout the analysis of the iterations. The Gaussian assumption on the components, and small enough number of iterations are crucial in this argument.

Notations: For two vectors $u, v \in \mathbb{R}^k$, the Hadamard product denoted by $*$ is defined as the entry-wise multiplication of vectors, i.e., $(u*v)_j := u_j v_j$ for $j \in [k]$. For a matrix A, let $P_{\perp_A}$ denote the projection operator to the subspace orthogonal to column span of A. For a subspace R, let $R^\perp$ denote the space orthogonal to it. Therefore, for a subspace R, the projection operator on the subspace orthogonal to R is equivalently denoted by $P_{R^\perp}$ or $P_{\perp_R}$. For a random matrix D, let $D|\{u = Dv\}$ denote the conditional distribution of D given linear constraints $u = Dv$.

Lemma 2.3 involves analyzing the dynamics of power iteration in (2.2) for 3rd order rank-k tensors. For the rank-k tensor in (2.18), the power iterative form $x \leftarrow \frac{T(I,x,x)}{\|T(I,x,x)\|}$ can be written as

$$x^{(t+1)} = \frac{A\left(A^\top x^{(t)}\right)^{*2}}{\left\|A\left(A^\top x^{(t)}\right)^{*2}\right\|}, \tag{2.20}$$

where the multilinear form in (1.3) is used. Here, $A = [a_1 \cdots a_k] \in \mathbb{R}^{d\times k}$ denotes the factor matrix, and for vector $y \in \mathbb{R}^k$, $y^{*2} := y * y \in \mathbb{R}^k$ represents the element-wise square of entries of y.

We consider the case where $a_i \sim \mathcal{N}(0, \frac{1}{d}I)$ are i.i.d. drawn and we analyze the evolution of the dynamics of the power update. As explained earlier, for a given initialization $x^{(1)}$, the update in the first step can be analyzed easily since A is independent of $x^{(1)}$. However, in subsequent steps, the updates $x^{(t)}$ are dependent on A, and it is no longer clear how to provide a tight bound on the evolution of $x^{(t)}$. In this work, we provide a careful analysis by controlling the amount of "correlation build-up" by exploiting the structure of Gaussian matrices under linear constraints.

This enables us to provide better guarantees for matrix A with Gaussian entries compared to general matrices A.

Intermediate update steps and variables: Before we proceed, we need to break down power update in (2.2) and introduce some intermediate update steps and variables as follows. Recall that $x^{(1)} \in \mathbb{R}^d$ denotes the initialization vector. Without loss of generality, let us analyze the convergence of power update to first component of rank-k tensor T denoted by a_1. Hence, let the first entry of $x^{(1)}$ denoted by $x_1^{(1)}$ be the maximum entry (in absolute value) of $x^{(1)}$, i.e., $x_1^{(1)} = \|x^{(1)}\|_\infty$. Let $B := [a_2\ a_3\ \cdots\ a_k] \in \mathbb{R}^{d\times(k-1)}$, and therefore $A = [a_1|B]$. We break the power update formula in (2.2) into a few steps by introducing intermediate variables $y^{(t)} \in \mathbb{R}^k$ and $\tilde{x}^{(t+1)} \in \mathbb{R}^d$ as

$$y^{(t)} := A^\top x^{(t)}, \quad \tilde{x}^{(t+1)} := A(y^{(t)})^{*2}.$$

Note that $\tilde{x}^{(t+1)}$ is the unnormalized version of $x^{(t+1)} := \tilde{x}^{(t+1)}/\|\tilde{x}^{(t+1)}\|$, i.e., $\tilde{x}^{(t+1)} := T(I, x^{(t)}, x^{(t)})$. Thus, we need to jointly analyze the dynamics of all variables $x^{(t)}$, $y^{(t)}$ and $(y^{(t)})^{*2}$. Define

$$X^{[t]} := \left[x^{(1)}|\dots|x^{(t)}\right], \quad Y^{[t]} := \left[y^{(1)}|\dots|y^{(t)}\right].$$

Matrix B is randomly drawn with i.i.d. Gaussian entries $B_{ij} \sim \mathcal{N}(0, \frac{1}{d})$. As the iterations proceed, we consider the following conditional distributions

$$B^{(t,1)} := B|\{X^{[t]}, Y^{[t]}\}, \quad B^{(t,2)} := B|\{X^{[t+1]}, Y^{[t]}\}. \tag{2.21}$$

Thus, $B^{(t,1)}$ is the conditional distribution of B at the middle of t^{th} iteration (before update step $\tilde{x}^{(t+1)} = A(y^{(t)})^{*2}$) and $B^{(t,2)}$ is the conditional distribution at the end of t^{th} iteration (after update step $\tilde{x}^{(t+1)} = A(y^{(t)})^{*2}$). By analyzing the above conditional distributions, we can characterize the left independent randomness in B.

2.6.1.1 Conditional Distributions

In order to characterize the conditional distribution of B under evolution of $x^{(t)}$ and $y^{(t)}$ in (2.21), we exploit the following basic fact (see [39] for proof).

Lemma 2.5 (Conditional distribution of Gaussian matrices under a linear constraint)**.** *Consider random matrix D with i.i.d. Gaussian entries $D_{ij} \sim \mathcal{N}(0, \sigma^2)$. Conditioned on $u = Dv$ with known vectors u and v, the matrix D is distributed as*

$$D|\{u = Dv\} \overset{(\mathrm{d})}{=} \frac{1}{\|v\|^2} uv^\top + \tilde{D}P_{\perp_v},$$

where random matrix $\tilde{D}$ is an independent copy of D with i.i.d. Gaussian entries $\tilde{D}_{ij} \sim \mathcal{N}(0, \sigma^2)$, and $P_{\perp_v}$ is the projection operator on to the subspace orthogonal to v.

We refer to $\tilde{D}P_{\perp_v}$ as the *residual* random matrix since it represents the remaining *randomness* left after conditioning. It is a random matrix whose rows are independent random vectors that are orthogonal to v, and the variance in each direction orthogonal to v is equal to σ^2.

The above Lemma can be exploited to characterize the conditional distribution of B introduced in (2.21). However, a naive direct application using the constraint $Y^{[t]} = A^\top X^{[t]}$ is not transparent for analysis. The reason is the evolution of $x^{(t)}$ and $y^{(t)}$ are themselves governed by the conditional distribution of B given previous iterations. Therefore, we need the following recursive version of Lemma 2.5.

Corollary 2.3 (Iterative conditioning)**.** *Consider random matrix D with i.i.d. Gaussian entries $D_{ij} \sim \mathcal{N}(0, \sigma^2)$, and let $F \overset{(\mathrm{d})}{=} P_{\perp_C} D P_{\perp_R}$ be the random Gaussian matrix whose columns are orthogonal to space C and rows are orthogonal to space R. Conditioned on the linear constraint $u = Dv$, where[11] $u \in C^\perp$, the matrix F is distributed as*

$$F|\{u = Dv\} \overset{(\mathrm{d})}{=} \frac{1}{\|(P_{\perp_R} v)\|^2} u(P_{\perp_R} v)^\top + P_{\perp_C} \tilde{D} P_{\perp_{\{R,v\}}},$$

where random matrix $\tilde{D}$ is an independent copy of D with i.i.d. Gaussian entries $\tilde{D}_{ij} \sim \mathcal{N}(0, \sigma^2)$.

[11] We need that $u \in C^\perp$, otherwise the event $u = Dv$ is impossible.

Thus, the *residual* random matrix $P_{\perp_C}\tilde{D}P_{\perp_{\{R,v\}}}$ is a random Gaussian matrix whose columns are orthogonal to C and rows are orthogonal to $\text{span}\{R, v\}$. The variance in any remaining dimension is equal to σ^2.

2.6.1.2 Form of Iterative Updates

Now we exploit the conditional distribution arguments proposed in the previous section to characterize the conditional distribution of B given the update variables x and y up to the current iteration; recall (2.21) where $B^{(t,1)}$ is the conditional distribution of B at the middle of t^{th} iteration and $B^{(t,2)}$ at the end of t^{th} iteration. Before that, we need to introduce some more intermediate variables.

Intermediate variables: We separate the first entry of y and $(y)^{*2}$ from the rest, i.e., we have

$$y_1^{(t)} = a_1^\top x^{(t)}, \quad y_{-1}^{(t)} = B^\top x^{(t)} \sim (B^{(t-1,2)})^\top x^{(t)},$$

where $y_{-1}^{(t)} \in \mathbb{R}^{k-1}$ denotes $y^{(t)} \in \mathbb{R}^k$ with the first entry removed. The update formula for $\tilde{x}^{(t+1)}$ can be also decomposed as

$$\tilde{x}^{(t+1)} = (y_1^{(t)})^2 a_1 + Bw^{(t)} \sim (y_1^{(t)})^2 a_1 + B^{(t,1)} w^{(t)},$$

where

$$w^{(t)} := (y_{-1}^{(t)})^{*2} \in \mathbb{R}^{k-1},$$

is the new intermediate variable in the power iterations. Let $B_{\text{res.}}^{(t,1)}$ and $B_{\text{res.}}^{(t,2)}$ denote the *residual* random matrices corresponding to $B^{(t,1)}$ and $B^{(t,2)}$ respectively, and

$$u^{(t+1)} := B_{\text{res.}}^{(t,1)} w^{(t)}, \quad v^{(t)} := (B_{\text{res.}}^{(t-1,2)})^\top x^{(t)},$$

where $u^{(t)} \in \mathbb{R}^d$ and $v^{(t)} \in \mathbb{R}^{k-1}$ are respectively the part of $x^{(t)}$ and $y_{-1}^{(t)}$ representing the residual randomness after conditioning on the previous iterations. We also summarize all variables and notations in Table B.1 in the Appendix which can be used as a reference throughout the chapter.

Finally we make the following observations.

Lemma 2.6 (Form of iterative updates)**.** *The conditional distribution of B at the middle of t^{th} iteration denoted by $B^{(t,1)}$ satisfies*

$$B^{(t,1)} \overset{(\mathrm{d})}{=} \sum_{i\in[t-1]} \frac{u^{(i+1)}(P_{\perp_{W^{[i-1]}}} w^{(i)})^\top}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|^2} + \sum_{i\in[t]} \frac{P_{\perp_{X^{[i-1]}}} x^{(i)}(v^{(i)})^\top}{\|P_{\perp_{X^{[i-1]}}} x^{(i)}\|^2} + B_{\text{res.}}^{(t,1)}, \tag{2.22}$$

$$B_{\text{res.}}^{(t,1)} \overset{(\mathrm{d})}{=} P_{\perp_{X^{[t]}}} \tilde{B} P_{\perp_{W^{[t-1]}}}, \tag{2.23}$$

where random matrix $\tilde{B}$ is an independent copy of B with i.i.d. Gaussian entries $\tilde{B}_{ij} \sim \mathcal{N}(0, \frac{1}{d})$. Similarly, the conditional distribution of B at the end of t^{th} iteration denoted by $B^{(t,2)}$ satisfies

$$B^{(t,2)} \overset{(\mathrm{d})}{=} \sum_{i\in[t]} \left(\frac{u^{(i+1)}(P_{\perp_{W^{[i-1]}}} w^{(i)})^\top}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|^2} + \frac{P_{\perp_{X^{[i-1]}}} x^{(i)}(v^{(i)})^\top}{\|P_{\perp_{X^{[i-1]}}} x^{(i)}\|^2} \right) + B_{\text{res.}}^{(t,2)}, \tag{2.24}$$

$$B_{\text{res.}}^{(t,2)} \overset{(\mathrm{d})}{=} P_{\perp_{X^{[t]}}} B' P_{\perp_{W^{[t]}}}, \tag{2.25}$$

where random matrix B' is an independent copy of B with i.i.d. Gaussian entries $B'_{ij} \sim \mathcal{N}(0, \frac{1}{d})$.

The lemma can be directly proved by applying the iterative conditioning argument in Corollary 2.3. See the detailed proof in the appendix.

2.6.1.3 Analysis of Iterative Updates

Lemma 2.6 characterizes the conditional distribution of B given the update variables x and y up to the current iteration; see (2.21) for the definition of conditional forms of B denoted by $B^{(t,1)}$ and $B^{(t,2)}$. Intuitively, when the number of iterations $t \ll d$, then the residual independent randomness left in $B^{(t,1)}$ and $B^{(t,2)}$ (respectively denoted by $B_{\text{res.}}^{(t,1)}$ and $B_{\text{res.}}^{(t,2)}$) characterized in Lemma 2.6 is

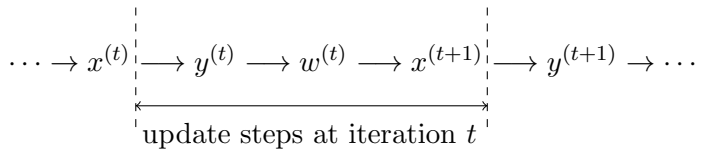


Figure 2.2: Flow of the power update algorithm stating intermediate steps. Iteration t for which the inductive step should be argued is also indicated.

large enough and we can exploit it to obtain tighter concentration bounds throughout the analysis of the iterations.

Note that the goal is to show that under $t \ll d$, the iterations $x^{(t)}$ converge to the true component with constant error, i.e., $|\langle x^{(t)}, a_1 \rangle| \geq 1 - \gamma$ for some constant $\gamma > 0$. If this already holds before iteration t we are done, and if it does not hold, next iteration is analyzed to finally achieve the goal. This analysis is done via *induction argument*. During the iterations, we maintain several invariants to analyze the dynamics of power update. The goal is to ensure progress in each iteration as in (2.26).

Induction hypothesis: The following are assumed at the beginning of the iteration t as induction hypothesis; see Figure 2.2 for the scope of inductive step.

1. **Length of Projection on x:**

$$\delta_t \leq \|P_{\perp_{X^{[t-1]}}} x^{(t)}\| \leq 1,$$

   where δ_t is of order $1/\operatorname{polylog} d$, and the value of δ_t only depends on t and $\log d$.

2. **Length of Projection on w:**

$$\delta'_{t-1} \frac{\sqrt{k}}{d} \leq \|P_{\perp_{W^{[t-2]}}} w^{(t-1)}\| \leq \Delta'_{t-1} \frac{\sqrt{k}}{d},$$
$$\|P_{\perp_{W^{[t-2]}}} w^{(t-1)}\|_\infty \leq \Delta'_{t-1} \frac{1}{d},$$

   where δ'_t is of order $1/\operatorname{polylog} d$ and Δ'_t is of order $\operatorname{polylog} d$. Both δ'_t and Δ'_t only depend on t and $\log d$.

3. **Progress:** [12]

$$|\langle a_1, x^{(t)}\rangle| \in [\delta_t^*, \Delta_t^*] d^{\beta 2^{t-1}} \frac{\sqrt{k}}{d}, \tag{2.26}$$
$$\langle a_1, P_{\perp_{X^{[t-1]}}} x^{(t)}\rangle \leq \Delta_t^* d^{\beta 2^{t-1}} \frac{\sqrt{k}}{d}.$$

4. **Norm of u,v:**

$$\frac{\delta_{t-1}}{2}\sqrt{\frac{k}{d}} \leq \|v^{(t-1)}\| \leq 2\sqrt{\frac{k}{d}},$$
$$\frac{\delta'_{t-1}}{2}\frac{\sqrt{k}}{d} \leq \|u^{(t)}\| \leq 2\Delta'_{t-1}\frac{\sqrt{k}}{d}.$$

The analysis for basis of induction and inductive step are provided in Appendix B.1.

2.6.2 Effect of noise in Theorem 2.3

Given rank-k random tensor T in (2.18), and a starting point $x^{(1)}$, our analysis in the noiseless setting shows that the tensor power iteration in (2.2) outputs a vector which will be close to a_j if $x^{(1)}$ has a large enough correlation with a_j.

Now suppose we are given noisy tensor $\hat{T} = T + E$ where E has some small norm. In this case where the noise is also present, we get a sequence $\hat{x}^{(t)} = x^{(t)} + \xi^{(t)}$ where $x^{(t)}$ is the component not incorporating any noise (as in previous section[13]), while $\xi^{(t)}$ represents the contribution of noise tensor E in the power iteration; see (2.27) below. We prove that $\xi^{(t)}$ is a very small noise that does not change our calculations stated in the following lemma.

Lemma 2.7 (Bounding norm of error)**.** *Suppose the spectral norm of the error tensor E is bounded as*

$$\delta_t \leq \|P_{\perp_{X^{[t-1]}}} x^{(t)}\| \leq 1,$$

[12] Note that although the bounds on $y_{-1}^{(t)}$ are argued at iteration t, the bound on the first entry of $y^{(t)}$ denoted by $y_1^{(t)} = \langle a_1, x^{(t)}\rangle$ is assumed here in the induction hypothesis at the end of iteration $t-1$.

[13] Note that there is a subtle difference between notation $x^{(t)}$ in the noiseless and noisy settings. In the noiseless setting, this vector is normalized, while in the noisy setting the whole vector $\hat{x}^{(t)} = x^{(t)} + \xi^{(t)}$ is normalized.

Then the noise vector $\xi^{(t)}$ at iteration t satisfies the ℓ_2 norm bound

$$\|\xi^{(t)}\| \leq \tilde{O}(d^{\beta 2^{t-1}}\epsilon).$$

Note that when t is the first number such that $d^{\beta 2^{t-1}} \geq d/\sqrt{k}$, we have $\|\xi^{(t)}\| = o(1)$.

Notice that since when $d^{\beta 2^{t-1}} \geq d/\sqrt{k}$, the main induction is already over and we know $x^{(t)}$ is constant close to the true component, and thus, the noise is always small.

Proof idea: We now provide an overview of ideas for proving the above lemma; see Appendix B.3 for the complete proof which is based on an induction argument. We first write the following recursion expanding the contribution of main signal and noise terms in the tensor power iteration as

$$\begin{aligned} x^{(t+1)} + \xi^{(t+1)} &= \text{Norm}\left(\hat{T}(x^{(t)} + \xi^{(t)}, x^{(t)} + \xi^{(t)}, I)\right) \\ &= \text{Norm}\left(T(x^{(t)}, x^{(t)}, I) + 2T(x^{(t)}, \xi^{(t)}, I) + T(\xi^{(t)}, \xi^{(t)}, I) + E(\hat{x}^{(t)}, \hat{x}^{(t)}, I)\right), \end{aligned} \tag{2.27}$$

where for vector v, we have $\text{Norm}(v) := v/\|v\|$, i.e., it normalizes the vector. The first term is the desired main signal and should have the largest norm, and the rest of the terms are the noise terms. The third term is of order $\|\xi^{(t)}\|^2$, and hence, it should be fine whenever we choose $\|E\|$ to be small enough. The last term is $O(\|E\|)$ and is the same for all iterations so that is also fine. The problematic term is the second term, whose norm if we bound naively is $2\|\xi^{(t)}\|$. However the normalization factor also contributes a factor of roughly $d/\sqrt{k}$, and thus, this term grows exponentially; it is still fine if we just do a constant number of iterations, but the exponent will depend on the number of iterations.

In order to solve this problem, and make sure that the amount of noise we can tolerate is independent of the number of iterations, we need a better way to bound the noise term $\xi^{(t)}$. The main problem here is we bound the norm of $\|T(x^{(t)}, \xi^{(t)}, I)\|$ by $\|T\|\|\xi^{(t)}\| \leq O(\xi^{(t)})$, by doing this we ignored the fact that $x^{(t)}$ is uncorrelated with the components in T. In order to get a tighter bound, we introduce another norm $\|\cdot\|_*$. Intuitively, the norm $\|\cdot\|_*$ captures the fact that x does not have

a high correlation with the components (except for the first component that x will converge to), and gives a better bound. In particular we have $\|T(x^{(t)}, \xi^{(t)}, I)\| \approx \frac{\sqrt{k}}{d}\|\xi^{(t)}\|_2$. Therefore, the normalization factor is compensated by the additional term $\frac{\sqrt{k}}{d}$. More concretely, this norm is defined as follows.

Definition 2.2 (Norm $\|\cdot\|_*$)**.** *Given a matrix $A = [a_1\ a_2\ \cdots\ a_k] \in \mathbb{R}^{d\times k}$, for any vector $u \in \mathbb{R}^d$, the norm $\|u\|_{A^*}$ is defined as*

$$\|u\|_{A^*} = \max_{i\in[k]} |\langle a_i, u\rangle|.$$

This norm satisfies a property shown in Lemma B.6 which enables us to argue that $\xi^{(t)}$ is small enough as stated in Lemma 2.7.

2.7 Experiments

In this section, we provide some synthetic experiments to evaluate the performance of Algorithm 1. Note that tensor power update in Algorithm 1 is the main step of our algorithm which is considered in this experiment. A random true tensor T is generated as follows. First, three components $A \in \mathbb{R}^{d\times k}$, $B \in \mathbb{R}^{d\times k}$, and $C \in \mathbb{R}^{d\times k}$ are randomly generated with i.i.d standard Gaussian entries. Then, the columns of these matrices are normalized where the normalization factors are aggregated as coefficients $w_j, j \in [k]$. From decomposition form in (1.5), tensor T is built through these random components. For each new initialization, $\widehat{a}^{(0)}$ and $\widehat{b}^{(0)}$ are randomly generated with i.i.d. standard Gaussian entries, and then normalized [14]. Initialization vector $\widehat{c}^{(0)}$ is generated through update formula in (2.6).

For each initialization $\tau \in [L]$, an alternative option of running the algorithm with a fixed number of iterations N is to stop the iterations based on some stopping criteria. In this experiment, we

[14] Drawing i.i.d. standard Gaussian entries and normalizing them is equivalent to drawing vectors uniformly from the d-dimensional unit sphere.

stop the iterations when the improvement in subsequent steps is small as

$$\max\left(\left\|\widehat{a}_\tau^{(t)}-\widehat{a}_\tau^{(t-1)}\right\|^2,\left\|\widehat{b}_\tau^{(t)}-\widehat{b}_\tau^{(t-1)}\right\|^2,\left\|\widehat{c}_\tau^{(t)}-\widehat{c}_\tau^{(t-1)}\right\|^2\right)\leq t_{\mathrm{S}},$$

where t_{S} is the stopping threshold. According to the bound in Theorem 2.4, we set

$$t_{\mathrm{S}}:=t_1(\log d)^2\frac{\sqrt{k}}{d}, \tag{2.28}$$

for some constant $t_1>0$.

Effect of size d and k

Algorithm 1 is applied to random tensors with $d=1000$ and $k=\{10,50,100,200,500,1000,2000\}$. The number of initializations is $L=2000$. The parameter t_1 in (3.20) is fixed as $t_1=1e-08$. Figure 3.3 and Table 3.1 illustrate the outputs of running experiments which is the average of 10 random runs.

Figure 3.3 depicts the ratio of recovered columns versus the number of initializations. Both horizontal and vertical axes are plotted in log-scale. We observe that it is much easier to recover the columns in the undercomplete settings ($k\leq d$), while it becomes harder when k increases. Linear start in Figure 3.3 suggests that recovering the first bunch of columns only needs polynomial number of initializations. For highly undercomplete settings like $d=1000$ and $k=10$, almost all columns are recovered in this linear phase. After this start, the concave part means that it needs many more initializations for recovering the next bunch of columns. As we go ahead, it becomes harder to recover true columns, which is intuitive.

Table 3.1 has the results from the experiments. Parameters k, stopping threshold t_{S}, and the average square error of the output, the average weight error and the average number of iterations are stated. The output averages are over several initializations and random runs. The square error

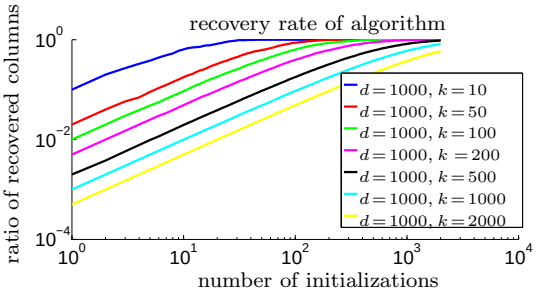


Figure 2.3: Ratio of recovered columns versus the number of initializations for $d = 1000$, and $k = \{10, 50, 100, 200, 500, 1000, 2000\}$. The number of initializations is $L = 2000$. The stopping parameter is set to $t_1 = 1e-08$. The figure is an average over 10 random runs.

is given by

$$\frac{1}{3}\left[\|a_j - \widehat{a}\|^2 + \left\|b_j - \widehat{b}\right\|^2 + \|c_j - \widehat{c}\|^2\right],$$

for the corresponding recovered j. The error in estimating the weights is defined as $|\widehat{w} - w_j|^2/w_j^2$ which is the square relative error of weight estimate. The number of iterations performed before stopping the algorithm is mentioned in the last column. We observe that by increasing k, all of these outputs are increased which means we get less accurate estimates with higher computation. This shows that recovering the overcomplete components is much harder. Note that by running the coordinate descent Algorithm 4, we can also remove this additional residual error left after the tensor power iteration step. Similar results and observations as above are seen when k is fixed and d is changed.

Running experiments with SVD initialization instead of random initialization yields nearly the same recovery rates, but with slightly smaller number of iterations. But, since the SVD computation is more expensive, in practice, it is desirable to initialize with random vectors. Our theoretical results for random initialization appear to be highly pessimistic compared to the efficient recovery results in our experiments. This suggests additional room for improving our theoretical guarantees under random initialization.

Table 2.1: Parameters and more outputs related to results of Figure 3.3. Note that $d = 1000$.

Parameters		Outputs		
k	t_{S}	avg. square error	avg. weight error	avg. # of iterations
10	1.51e-08	1.03e-05	9.75e-09	7.71
50	3.37e-08	5.54e-05	6.69e-08	8.53
100	4.77e-08	1.08e-04	1.51e-07	8.81
200	6.75e-08	2.07e-04	3.41e-07	9.09
500	1.07e-07	5.09e-04	1.14e-06	9.52
1000	1.51e-07	1.01e-03	3.40e-06	10.01
2000	2.13e-07	2.00e-03	1.12e-05	10.69

Chapter 3

Learning Overcomplete Representations Using Tensor Methods

In this chapter, we provide guarantees for learning latent variable models and latent representations emphasizing on the overcomplete regime, where the dimensionality of the latent space exceeds the observed dimensionality. In particular, we consider multiview mixtures, ICA, and sparse coding models. Our main tool is a new algorithm for tensor decomposition that works in the overcomplete regime. We analyzed the performance of this algorithm in the previous chapter. In this chapter, we recap how learning different latent variable models can be formulated as a tensor decomposition algorithm. By proving new tensor concentration bounds, we are able to provide sample complexity results for learning these models by tensor methods.

In the semi-supervised setting, we exploit label information to get a rough estimate of the model parameters, and then refine it using the tensor method on unlabeled samples. We establish learning guarantees when the number of components scales as $k = o(d^{p/2})$, where d is the observed dimension, and p is the order of the observed moment employed in the tensor method (usually $p = 3, 4$). In the unsupervised setting, a simple initialization algorithm based on SVD of the tensor slices is proposed,

and the guarantees are provided under the stricter condition that $k \leq \beta d$ (where constant β can be larger than 1). We also provide tight sample complexity bounds through novel covering arguments.

It is often useful to incorporate latent variables in any modeling framework. Latent variables can capture the effect of hidden causes which are not directly observed. Learning these hidden factors is central to many applications, e.g., identifying the latent diseases through observed symptoms, identifying the latent communities through observed social ties, and so on. Moreover, latent variable models (LVMs) can provide an efficient representation of the observed data, and learning these representations can lead to improved performance on various tasks such as classification. The recent performance gains in domains such as speech and computer vision can be largely attributed to efficient representation learning [40]. Moreover, it has been shown that learning overcomplete representations is crucial to achieving these impressive gains [58].

In an overcomplete representation, the dimensionality of the latent space exceeds the observed dimensionality. Overcomplete representations are known to be more robust to noise, and can provide greater flexibility in modeling [119]. Although overcomplete representations have led to huge performance gains in practice, theoretical guarantees for learning are mostly lacking. In many domains, we face the challenging task of unsupervised or semi-supervised learning, since it is expensive to obtain labeled samples and we typically have access to a large number of unlabeled samples, e.g. [58, 117]. Therefore, it is imperative to develop novel guaranteed methods for efficient unsupervised/semi-supervised learning of overcomplete models.

In this chapter, we bridge the gap between theory and practice, and establish that a wide range of overcomplete LVMs can be learned efficiently through simple spectral learning techniques. We perform spectral decomposition of the higher order moment tensors (estimated using unlabeled samples) to obtain the model parameters. A recent line of work has shown that tensor decompositions can be employed for unsupervised learning of a wide range of LVMs, e.g., independent components analysis [69], topic models, Gaussian mixtures, hidden Markov models [15], network community models [12], and so on. It involves decomposition of a multivariate moment tensor, and is guaranteed to provide a consistent estimate of the model parameters. The sample and computational requirements are only a low order polynomial in the latent dimensionality for the tensor

method [15, 147]. However, a major drawback behind these works is that they mostly consider the undercomplete setting, where the latent dimensionality cannot exceed the observed dimensionality.

In practice, the tensor decomposition techniques have been shown to be effective in a number of applications such as blind source separation [60], computer vision [155], contrastive topic modeling [162], and community detection [97], where the tensor approach is shown to be orders of magnitude faster than existing techniques such as the stochastic variational approach.

In this work, we establish guarantees for tensor decomposition in learning overcomplete LVMs, such as multiview mixtures, independent component analysis, Gaussian mixtures and sparse coding models. Note that learning general overcomplete models is ill-posed since the latent dimensionality exceeds the observed dimensionality. We impose a natural incoherence condition on the components, which can be viewed as a *soft orthogonality* constraint, which limits the redundancy among the components. We establish that this constraint not only makes learning well-posed but also enables efficient learning through tensor methods. Incoherence constraints are natural in the overcomplete regime, and have been considered before, e.g., in compressed sensing [74], independent component analysis [117], and sparse coding [27, 3].

3.1 Summary of Results

In this chapter, we provide semi-supervised and unsupervised learning guarantees for LVMs such as multiview mixtures, ICA and sparse coding models. Our algorithm is based on method of moments, and employs a tensor decomposition algorithm for learning. Under the semi-supervised setting, we establish that highly overcomplete models can be learned efficiently through the tensor decomposition method. The moment tensors are constructed using unlabeled samples, and the labeled samples are used to provide a rough initialization to the tensor decomposition algorithm. In the unsupervised setting, we propose a simple initialization strategy for the tensor method, and can handle mildly overcomplete models. In both settings we provide tight sample complexity bounds through novel covering arguments.

3.1.1 Learning Multiview Mixture Model

In the multiview mixtures model, given the hidden mixture component, each observation (view) is independently drawn with some unknown mean parameter and noise distribution around that mean; see Section 3.3 for details. The goal is to estimate the conditional mean parameters. In this setting, we assume reasonable property on noise, and for brevity, we consider the "low" noise regime (where the norm of noise is of the same order as that of the component means).

In the *semi-supervised* setting, we use labeled samples to initialize the tensor decomposition algorithm, and provide the following recovery guarantee.

Theorem 3.1 (Semi-supervised learning of multiview mixtures model: informal)**.** *Let k be the number of mixture components, and d be the observed dimensionality, and suppose $k \leq o(d^{1.5})$. We show that having* $\text{polylog}(d, k)$ *number of labeled samples for each label, and $n \geq \tilde{\Omega}(k)$ number of unlabeled samples are sufficient to consistently estimate the model parameters.*

See Theorem 3.7 for the formal statement of this result. Thus, for recovering each rank-1 component, we need far less number of labeled samples compared to the number of unlabeled samples required. Note that in most applications, labeled samples are expensive/hard to obtain, while many more unlabeled samples are easily available, e.g., see [117, 57]. Furthermore, note that the unlabeled sample complexity is the *minimax* bound up to polylog factors.

We also provide *unsupervised* learning guarantees when no label is available. Here, the initialization is performing by the SVD-based method stated in the previous section. This imposes additional conditions on rank and sample complexity as follows.

Theorem 3.2 (Unsupervised learning of multiview mixtures model: informal)**.** *Suppose the number of unlabeled samples n satisfies $n \geq \tilde{\Omega}(kd)$. If $k \leq \beta d$ (for arbitrary constant $\beta > 1$), then the model parameters can be learned using a polynomial number of initializations scaled as k^{β^2}.*

See Theorem 3.8 for the formal statement of this result. This result is an improvement over existing results since we do not have dependence on the condition number of the component means and in addition, we can handle overcomplete models.

3.1.2 Learning ICA and Sparse ICA (Dictionary Learning) Models

We also provide semi-supervised and unsupervised learning guarantees for *Independent Component Analysis* (ICA). By semi-supervised setting in ICA, we mean some prior information is available which provides good initializations for the tensor decomposition algorithm. In the *semi-supervised* setting, we show that when the number of components $k = \Theta(d^2)/\operatorname{polylog}(d)$, the ICA model can be efficiently learned from fourth order moments with $n \geq \tilde{\Omega}(k^{2.5})$ number of unlabeled samples. In the *unsupervised* setting, we show that when $k \leq \beta d$ (for arbitrary constant $\beta > 1$), the ICA model can be learned with number of samples scaling as $n \geq \tilde{\Omega}(k^3)$ in k^{β^2} number of initializations.

We also provide learning results for the *sparse coding* model, when the mixing coefficients are independently drawn from a Bernoulli-Gaussian distribution and the dictionary satisfies some deterministic conditions (see Appendix A.2 and (RIP) in Section 3.3.1). Notice this corresponds to a sparse ICA model since the hidden coefficients are independent.

Theorem 3.3 (Learning (sparse) ICA: informal)**.** *We can efficiently estimate the dictionary in the (sparse) ICA model under the following conditions. Let s be the expected sparsity of the hidden coefficients. In the* semi-supervised *setting (where prior information provides us good initialization), we need the number of components to be bounded by $k = o(d^2)$, and unlabeled sample complexity satisfies $n \geq \tilde{\Omega}(\max\{sk, s^2k^2/d^3\})$. In the* unsupervised *setting, we need $k = \Theta(d)$, and $n \geq \tilde{\Omega}(k^2 s)$.*

In the special case when s is a constant, the sample complexity is akin to learning multiview models, and when $s = \Theta(k)$, it is akin to learning the "dense" ICA model. Thus, the sparse coding model bridges the range of models between multiview mixtures model and ICA. See Theorem 3.12 for the formal statement of above result on learning sparse ICA.

3.2 Related Works

Several latent variable models can be learned through tensor decomposition including independent component analysis [69], topic models, Gaussian mixtures, hidden Markov models [15] and network community models [12]. In the undercomplete setting, Anandkumar et al. [15] analyze robust

tensor power iteration for learning LVMs, and Song et al. [147] extend analysis to the nonparametric setting. These works require the tensor factors to have full column rank, which rules out overcomplete models. Moreover, they require whitening the input data, and the sample complexity depends on the condition number of the factor matrices. For instance, when $k = d$, for random factor matrices, the previous tensor approaches in [147, 11] have a sample complexity of $\tilde{\Omega}(k^{6.5})$. Our result can be also extended to learning mixtures of spherical Gaussians, where we have better sample complexity than the work by Hsu and Kakade [94] (we have $\tilde{\Omega}(d^2)$ instead of their $\tilde{\Omega}(d^3)$ when $k = d$). Note that this comparison is in the low noise regime (where the norm of noise is of the same order as that of the component means). Thus, we provide the best known sample bounds for semi-supervised and unsupervised learning of multiview mixtures model in the overcomplete setting, assuming incoherent components.

In general, learning overcomplete models is challenging, and they may not even be identifiable in general. The FOOBI procedure by De Lathauwer et al. [69] shows that a polynomial-time procedure can recover the components of ICA model (with *generic* factors) when $k = O(d^2)$, where the moment is fourth order. However, the procedure does not work for third-order overcomplete tensors. For the fifth order tensor, Goyal et al. [80], Bhaskara et al. [41] perform simultaneous diagonalization on the matricized versions of random slices of the tensor and provide careful perturbation analysis. But, this procedure cannot handle the same level of overcompleteness as FOOBI. In addition, Goyal et al. [80] provide stronger results for ICA, where the tensor slices can be obtained in the Fourier domain. Given 4th order tensor, they need poly(k^4) number of unlabeled samples for learning ICA (where the poly factor is not explicitly characterized), while we only need $\tilde{\Omega}(k^{2.5})$ (when $k = \Theta(d^2)/\operatorname{polylog}(d)$).

Learning mixture of Gaussians: Here, we provide a subset of related works studying learning mixture of Gaussians which are more comparable with our result. For a more detailed list of these works, see Anandkumar et al. [18], Hsu and Kakade [95]. The problem of learning mixture of Gaussians dates back to the work by Pearson [133]. They propose a moment-based technique that involves solving systems of multivariate polynomials which is in general challenging in both computational and statistical sense. Recently, lots of studies on learning Gaussian mixture models

have been done improving both aspects which can be divided to two main classes: distance-based and spectral methods.

Distance-based methods impose separation condition on the mean vectors showing that under enough separation the parameters can be estimated. Among such approaches, we can mention Dasgupta [65], Vempala and Wang [156], Arora and Kannan [29]. As discussed in the summary of results, these results work even if $k > d^{1.5}$ as long as the separation condition between means is satisfied, but our work can tolerate higher level of noise in the regime of $k = o(d^{1.5})$ with polynomial computational complexity. The guarantees in [156] also work in the high noise regime but need higher computational complexity as polynomial in $k^{O(k)}$ and d.

In the spectral approaches, the observed moments are constructed and the spectral decomposition of the observed moments are performed to recover the parameters [104, 10, 21]. Kalai et al. [104] analyze the problem of learning mixture of two general Gaussians and provide algorithm with high order polynomial sample and computational complexity. Note that in general, the complexity of such methods grow exponentially with the number of components without further assumptions [127]. Hsu and Kakade [95] provide a spectral algorithm under non-degeneracy conditions on the mean vectors and providing guarantees with polynomial sample complexity depending on the condition number of the moment matrices. Anandkumar et al. [21] perform tensor power iteration on the third order moment tensor to recover the mean vectors in the overcomplete regime as long as $k = o(d^{1.5})$, but need very good initialization vector having constant correlation with the true mean vector. Here, we improve the correlation level required for convergence.

More discussions on related works is provided in Appendix A.1.

3.3 Tensor Decomposition for Learning Latent Variable Models

In this section, we discuss that the problem of learning several latent variable models reduces to the tensor decomposition problem. We show that the observed moment of the latent variable models can be written in a CP tensor decomposition form when appropriate modifications are

performed. This is done for multiview linear mixtures model, spherical Gaussian mixtures and ICA (Independent Component Analysis). For a more detailed discussion on the connection between observed moments of LVMs and tensor decomposition, see Section 3 in Anandkumar et al. [15].

Therefore, an efficient tensor decomposition method leads to efficient learning procedure for a wide range of latent variable models. We exploit the algorithm and analysis in Chapter 2 for learning latent variable models providing sample complexity results in the subsequent sections. Note that the sample complexity guarantees are argued through tensor concentration bounds proposed in Section 3.4.

3.3.1 Multiview linear mixtures model

Consider a multiview linear mixtures model as in Figure 3.1 with k components and $p \geq 3$ views. Throughout the chapter, we assume $p = 3$ for simplicity, while the results can be also extended to higher-order. Suppose that hidden variable $h \in [k]$ is a discrete categorical random variable with $\Pr[h = j] = w_j, j \in [k]$. The variables (views) $x_l \in \mathbb{R}^d$ are conditionally independent given the k-categorical latent variable $h \in [k]$, and the conditional means are

$$\mathbb{E}[x_1|h] = a_h, \quad \mathbb{E}[x_2|h] = b_h, \quad \mathbb{E}[x_3|h] = c_h, \tag{3.1}$$

where $A := [a_1\ a_2\ \cdots\ a_k] \in \mathbb{R}^{d\times k}$ denotes the *factor matrix* and B, C are similarly defined. The goal of the learning problem is to recover the parameters of the model (factor matrices) A, B, and C given observations.

For this model, the third order observed moment has the form (See Anandkumar et al. 15)

$$\mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \sum_{j\in[k]} w_j a_j \otimes b_j \otimes c_j. \tag{3.2}$$

The decomposition in (3.2) is referred to as the CP decomposition [49], and k denotes the CP tensor rank. Hence, given third order observed moment, the unsupervised learning problem (recovering factor matrices A, B, and C) reduces to computing a tensor decomposition as in (3.2).

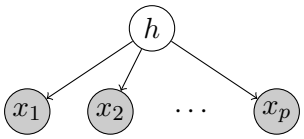


Figure 3.1: Multi-view mixtures model

In addition, suppose that given hidden state h, the observed variables $x_l \in \mathbb{R}^d$ have conditional distributions as

$$x_1|h \sim a_h + \zeta\sqrt{d} \cdot \varepsilon_A, \quad x_2|h \sim b_h + \zeta\sqrt{d} \cdot \varepsilon_B, \quad x_3|h \sim c_h + \zeta\sqrt{d} \cdot \varepsilon_C,$$

where $\varepsilon_A, \varepsilon_B, \varepsilon_C \in \mathbb{R}^d$ are independent random vectors with zero mean and covariance $\frac{1}{d}I_d$, and ζ^2 is a scalar denoting the variance of each entry. We also assume that noise vectors $\varepsilon_A, \varepsilon_B, \varepsilon_C$ are independent of hidden vector h. In addition, let all the vectors $a_h, b_h, c_h, h \in [k]$, have unit ℓ_2 norm. Furthermore, since w_j's are the mixture probabilities, for simplicity we consider $w_j = \Theta(1/k), j \in [k]$. We call this model $\mathcal{S}$.

When $\zeta^2 = \Theta(1/d)$, the norm of the noise is roughly the same as the norm of the components. We call this the *low noise regime*. When $\zeta^2 = \Theta(1)$, the norm of noise in *every dimension* is roughly the same as the norm of the components. We call this the *high noise regime*.

3.3.2 Spherical Gaussian mixtures

Consider a mixture of k different Gaussian distributions with spherical covariances. Let $w_j, j \in [k]$ denote the proportion for choosing each mixture. For each Gaussian component $j \in [k]$, $a_j \in \mathbb{R}^d$ is the mean, and $\zeta_i^2 I$ is the spherical covariance. For simplicity, we restrict to the case where all the components have the same spherical variance, i.e., $\zeta_1^2 = \zeta_2^2 = \cdots = \zeta_k^2 = \zeta^2$. The generalization is discussed in Hsu and Kakade [94]. In addition, in order to generalize the learning result to the overcomplete setting, we assume that variance parameter ζ^2 is known (see Remark 1 for more discussions). The following lemma shows that the problem of estimating parameters of this mixture model can be formulated as a tensor decomposition problem. This is a special case of Theorem 1 in Hsu and Kakade [94] where we assume the variance parameter is known.

Lemma 3.1 (Hsu and Kakade 94)**.** *If*

$$M_3 := \mathbb{E}[x \otimes x \otimes x] - \zeta^2 \sum_{i \in [d]} \left(\mathbb{E}[x] \otimes e_i \otimes e_i + e_i \otimes \mathbb{E}[x] \otimes e_i + e_i \otimes e_i \otimes \mathbb{E}[x] \right), \tag{3.3}$$

then

$$M_3 = \sum_{j \in [k]} w_j a_j \otimes a_j \otimes a_j.$$

In order to provide the learning guarantee, we define the following empirical estimates. Let $\widehat{\mathcal{M}}_3$, $\widehat{\mathcal{M}}_2$, and $\widehat{\mathcal{M}}_1$ respectively denote the empirical estimates of the raw moments $\mathbb{E}[x \otimes x \otimes x]$, $\mathbb{E}[x \otimes x]$, and $\mathbb{E}[x]$. Then, the empirical estimate of the third order modified moment in (3.3) is

$$\widehat{M}_3 := \widehat{\mathcal{M}}_3 - \zeta^2 \sum_{i \in [d]} \left(\widehat{\mathcal{M}}_1 \otimes e_i \otimes e_i + e_i \otimes \widehat{\mathcal{M}}_1 \otimes e_i + e_i \otimes e_i \otimes \widehat{\mathcal{M}}_1 \right). \tag{3.4}$$

Remark 1 (Variance parameter estimation). Notice that we assume variance ζ^2 is known in order to generalize the learning result to the overcomplete setting. Since ζ is a scalar parameter, it is reasonable to try different values of ζ till we get a good reconstruction. On the other hand, in the undercomplete setting, variance ζ^2 can be also estimated as proposed in [94], where estimate $\hat{\zeta}^2$ is the k-th largest eigenvalue of the empirical covariance matrix $\widehat{\mathcal{M}}_2 - \widehat{\mathcal{M}}_1 \widehat{\mathcal{M}}_1^\top$.

3.3.3 Independent component analysis (ICA)

In the standard ICA model [59, 48, 98, 61], random independent latent signals are linearly mixed and perturbed with noise to generate the observations. Let $h \in \mathbb{R}^k$ be a random latent signal, where its coordinates are independent, $A \in \mathbb{R}^{d \times k}$ be the mixing matrix, and $z \in \mathbb{R}^d$ be the Gaussian noise. In addition, h and z are also independent. Then, the observed random vector is

$$x = Ah + z.$$

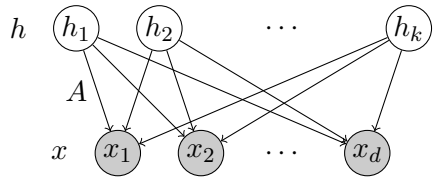


Figure 3.2: Graphical representation of ICA (Independent Component Analysis) model $x = Ah$, where the coordinates of h are independent.

Figure 3.2 depicts a graphical representation of the ICA model where the coordinates of h are independent.

The following lemma shows that the problem of estimating parameters of the ICA model can be formulated as a tensor decomposition problem.

Lemma 3.2 (Comon and Jutten 61)**.** *Define*

$$M_4 := \mathbb{E}[x \otimes x \otimes x \otimes x] - T, \tag{3.5}$$

where $T \in \mathbb{R}^{d \times d \times d \times d}$ *is the fourth order tensor with*

$$T_{i_1,i_2,i_3,i_4} := \mathbb{E}[x_{i_1}x_{i_2}]\mathbb{E}[x_{i_3}x_{i_4}] + \mathbb{E}[x_{i_1}x_{i_3}]\mathbb{E}[x_{i_2}x_{i_4}] + \mathbb{E}[x_{i_1}x_{i_4}]\mathbb{E}[x_{i_2}x_{i_3}], \quad i_1, i_2, i_3, i_4 \in [d]. \tag{3.6}$$

Let $\kappa_j := \mathbb{E}[h_j^4] - 3\mathbb{E}^2[h_j^2]$, $j \in [k]$. *Then, we have*

$$M_4 = \sum_{j \in [k]} \kappa_j a_j \otimes a_j \otimes a_j \otimes a_j. \tag{3.7}$$

See [94] for a proof of this theorem in this form. Let $\widehat{M_4}$ be the empirical estimate of M_4 given n samples.

Sparse ICA

We also consider the sparse ICA model, which is the ICA with the additional constraint that the hidden vector h is sparse.

This is related to the dictionary learning or sparse coding model $x = Ah$ where the observations $x \in \mathbb{R}^d$ are sparse combination of dictionary atoms $a_j \in \mathbb{R}^d, j \in [k]$ through sparse vector $h \in \mathbb{R}^k$. If in addition, the coordinates of h are random and independent, the dictionary learning model is the same as the sparse ICA model. Others have studied the general sparse coding problem which are briefly mentioned in the related works section.

3.4 Tensor Concentration Bounds

In this section, we provide tensor concentration results for the proposed latent variable models. For each LVM, consider the higher-order observed moment (tensor) described in Section 3.3. The tensor concentration result bounds the spectral norm of error between the true moment tensor and its empirical estimate given n samples.

3.4.1 Multiview linear mixtures model

For the multiview linear mixtures model, we provide the tensor concentration result for the 3rd order observed moment in (3.2).

Consider the multiview linear mixtures model described in Section 3.3.1 denoted as model $\mathcal{S}$. Let $x_1^i, x_2^i, x_3^i, i \in [n]$, denote n samples of views x_1, x_2, x_3, respectively. Since the main focus is on recovering the components, we bound the spectral norm of difference between the empirical tensor estimate

$$\hat{T} := \frac{1}{n} \sum_{i=1}^{n} x_1^i \otimes x_2^i \otimes x_3^i,$$

and

$$\tilde{T} := \mathbb{E}\big[x_1 \otimes x_2 \otimes x_3 | h_i, i \in [n]\big] = \frac{1}{n}\sum_{i=1}^{n} (a_{h_i}) \otimes (b_{h_i}) \otimes (c_{h_i}),$$

where the expectation is conditioned on the choice of hidden states for n samples, and taken over the randomness of noise. Here, $h_i \in [k]$ denotes the hidden state for sample $i \in [n]$. Notice that tensor $\tilde{T}$ has the same form as true tensor T in (3.2) where

$$\tilde{T} = \sum_{j \in [k]} \tilde{w}_j a_j \otimes b_j \otimes c_j.$$

Here $\tilde{w}_j, j \in [k]$ are the empirical frequencies of different hidden states $h \in [k]$. It is easy to see that if $n \geq \Omega\left(\frac{\log k}{w_{\min}}\right)$, then all the empirical frequencies $\tilde{w}_j$ are within $[w_j/2, 2w_j]$. Therefore, tensor decomposition of $\tilde{T}$ has the same eigenvectors and similar eigenvalues as the true expectation (over both the noise and the hidden variables), and hence, it suffices to bound $\|\hat{T} - \tilde{T}\|$ provided as follows.

Theorem 3.4 (Tensor concentration bound for multiview linear mixtures model)**.** *Consider n samples $\{(x_1^i, x_2^i, x_3^i), i \in [n]\}$ from the multiview linear mixtures model $\mathcal{S}$ with corresponding hidden states $\{h_i, i \in [n]\}$. Assume matrices $A^\top$, $B^\top$ and $C^\top$ have $2 \to 3$ norm bounded by $O(1)$, and noise matrices E_A, E_B and E_C defined in* (3.9) *satisfy the RIP condition in (RIP) (see Remark 3 for details on RIP condition). For $\hat{T}$ and $\tilde{T}$ as above, if $n = \mathrm{poly}(d)$, we have with high probability (over the choice of hidden state h and the noise)*

$$\|\hat{T} - \tilde{T}\| \leq \tilde{O}\left(\zeta\left(\frac{\sqrt{d}}{n} + \sqrt{w_{\max}\frac{d}{n}}\right) + \zeta^2\left(\frac{d}{n} + \sqrt{w_{\max}\frac{d^{1.5}}{n}}\right) + \zeta^3\left(\frac{d^{1.5}}{n} + \sqrt{\frac{d}{n}}\right)\right).$$

See the proof in Appendix C.2.1. The main ideas are described later in this section.

The above bound holds for any level of noise, but in each specific regime of noise, one of the terms is dominant and the bound is simplified. We now provide the bound for the high noise $\zeta^2 = \Theta(1)$ and low noise $\zeta^2 = \Theta(1/d)$ regimes which were introduced in Section 3.3.1. In the high noise regime $\zeta^2 = \Theta(1)$, the term $\zeta^3\sqrt{\frac{d}{n}}$ in Theorem 3.4 is dominant, and in the low noise regime

$\zeta^2 = \Theta(1/d)$, the term $\zeta\sqrt{w_{\max}\frac{d}{n}}$ in Theorem 3.4 is dominant. This concentration bound is later used in Section 3.6 to provide sample complexity guarantees for learning multiview linear mixtures model.

Remark 2 (Application of Theorem 3.4 to whitening-based approaches). In the undercomplete setting, a guaranteed approach for tensor decomposition is to first orthogonalize the tensor through the *whitening* step, and then perform the orthogonal tensor eigen-decomposition through the power method [15]. The whitening step leads to dependency to the condition number in the sample complexity result. Applying the proposed tensor concentration bound in Theorem 3.4 to this approach, we get similar dependency to the condition number, but better dependency in the dimension d. This improvement comes at the cost of additional bounded $2 \to 3$ norm condition on the factor matrices.

Concretely, following the analysis in [15, 147], we have the error in recovery (up to permutation) as

$$\|\hat{a}_i - a_i\| \leq \frac{32\sqrt{2}\epsilon_{\text{triples}}}{\sigma^3_{\min} w^{1.5}_{\min}} + \frac{512\epsilon^3_{\text{pairs}}}{\sigma^3_{\min} w^{1.5}_{\min}}, \tag{3.8}$$

where $\epsilon_{\text{triples}} := \|\hat{T} - \tilde{T}\|$ is the error in estimating the third order moment, ϵ_{pairs} is the error in estimating the second order moments and $\sigma_{\min}$ is the k^{th} singular value of the factor matrices. While the ϵ_{pairs} can be obtained by matrix Bernstein's bounds as before (e.g. see [10]), we have an improved bound for $\epsilon_{\text{triples}}$ from Theorem 3.4, compared to previous results. Note that the first term corresponding to $\epsilon_{\text{triples}}$ is the dominant one and we improve its scaling.

Remark 3 (RIP property). Given n samples for the model $\mathcal{S}$ proposed in Section 3.3.1, define noise matrix

$$E_A := [\varepsilon^1_A, \varepsilon^2_A, \ldots, \varepsilon^n_A] \in \mathbb{R}^{d \times n}, \tag{3.9}$$

where $\varepsilon^i_A \in \mathbb{R}^d$ is the i-th sample of noise vector ε_A. E_B and E_C are similarly defined. These matrices need to satisfy the RIP property as follows which is adapted from Candes and Tao [47].

(RIP) Matrix $E \in \mathbb{R}^{d \times n}$ satisfies a weak RIP condition such that for any subset of $O\left(\frac{d}{\log^2 d}\right)$ number of columns, the spectral norm of E restricted to those columns is bounded by 2.

It is known that when $n = \text{poly}(d)$, the above condition is satisfied with high probability for many random models such as when the entries are i.i.d. zero mean Gaussian or Bernoulli random variables.

Proof ideas: The basic idea for proving the concentration result in Theorem 3.4 is an ε-net argument. We construct an ε-net and then show that with high probability the norm of error tensor is bounded for every vector in the ε-net.

In some cases even a usual ε-net of size $e^{O(d)}$ is good enough. But, in many other cases the usual ε-net construction does not provide a useful result since the failure probability is not small enough, and the union bound argument over all vectors in the ε-net fails (or incurs additional polynomial factors in the sample complexity result). In particular, for a vector with high correlation with the data, we get a worse concentration bound. But, the key observation is that there can not be too many vectors that have high correlation with the data. Therefore, for each fixed vector in the ε-net, we partition the terms in the error into two sets; one set corresponds to the small terms (where the vector is not highly correlated with the data) and the other set corresponds to the large terms. For the small terms, the usual ε-net argument still works. For the large terms, we show that the number of such terms is limited. This is done either by RIP property of the noise matrices or by the bounded $2 \to 3$ norm of factor matrices $A^\top$, $B^\top$ and $C^\top$. See the proofs of Claims 12-14 for more details. This partitioning argument is inspired by the entropy-concentration trade-off proposed in [138]; however, here we have a finer partitioning into several sets, while in [138] the partitioning is done into only two sets.

Spherical Gaussian mixtures: Similar tensor concentration bound as above holds for the spherical Gaussian mixtures model with exploiting symmetrization trick as follows. In the spherical Gaussian mixtures model, the modified higher order moment (tensor) in (3.3) is symmetric, and hence noise matrices E_A, E_B and E_C are all the same. This can cause a problem because some square terms in the error tensor are not zero mean and we need to show their concentration around

the mean. The well-known *symmetrization technique* can be exploited here where we draw two independent set of samples, and show the difference between the two is with high probability small. This technique is widely applied to show concentration around the median, and in all our cases the median is very close to the mean.

3.4.2 ICA and sparse ICA

For the ICA model, we provide the tensor concentration result for the modified 4th order observed moment (tensor) in (3.5) in both dense and sparse cases.

Theorem 3.5 (Tensor concentration bound for ICA)**.** *Consider n samples $x^i = Ah^i, i \in [n]$ from the ICA model with mixing matrix $A \in \mathbb{R}^{d\times k}$. Suppose $\|A\| \leq O(1+\sqrt{k/d})$ and the entries of $h \in \mathbb{R}^k$ are independent subgaussian variables with $\mathbb{E}[h_j^2] = 1$ and constant nonzero 4th order cumulant. For the 4th order cumulant M_4 in* (3.5) *and its empirical estimate $\widehat{M}_4$, if $n \geq d$, we have with high probability*

$$\|\widehat{M}_4 - M_4\| \leq \tilde{O}\left(\frac{m^2}{n} + \sqrt{\frac{m^4}{d^3 n}}\right), \quad m := \max(d,k).$$

See the proof in Appendix C.2.2. We have an improved bound for the sparse ICA setting as follows.

Theorem 3.6 (Tensor concentration bound for sparse overcomplete ICA)**.** *In the ICA model $x = Ah$, suppose $h_j = s_j g_j$ where s_j's are i.i.d. Bernoulli random variables with $\Pr[s_j = 1] = s/k$, and g_j's are independent 1-subgaussian random variables. Consider n independent samples $x^i = Ah^i, i \in [n]$, where each h^i is distributed as h. Suppose A satisfies (RIP) property (see Remark 3 for details on RIP condition). For the 4th order cumulant M_4 in* (3.5) *and its empirical estimate $\widehat{M}_4$, if $n, k \geq d$, we have with high probability*

$$\|\widehat{M}_4 - M_4\| \leq \tilde{O}\left(\frac{s^2}{n} + \sqrt{\frac{s^4}{d^3 n}}\right).$$

See the proof in Appendix C.2.3.

Dependence on k: It may seem counter-intuitive that the bound in Theorem 3.6 does not depend on k. The dependency on k is actually in the expectation where the expected tensor $\mathbb{E}\big[x^{\otimes 4}\big]$ in M_4 is close to $\frac{s}{k}\sum_{j\in[k]} a_j^{\otimes 4}$. We typically require the deviation to be less than the expected value.

Proof ideas: The proof ideas are similar to the multiview mixtures model where we provide ε-net arguments and partition the terms to small and large ones. In addition, for the ICA model, we exploit the subgaussian property of h_j's to provide concentration bound for the summation of subgaussian random variables raised to the 4th power (see Claim 15). This implies the concentration bound for the 4th order term $\mathbb{E}[x^{\otimes 4}]$ in M_4 (see Claim 16). For the 2nd order term T in M_4, the bound is argued using Matrix Bernstein's inequality (see Claim 17). For the sparse ICA model, the RIP property of A is exploited to bound the size of intersection between the support of (partitioned) vectors in the ε-net and the support of sparse vectors h^i (see Claim 18).

3.5 Learning Algorithm

We exploit the tensor decomposition algorithm proposed in Section 2.3 to learn the latent variable models. The only difference here is we use the label information for initialization in the semi-supervised setting. More concretely, the initialization in Algorithm 1 is performed as follows:

- Semi-supervised setting: label information is exploited. See equation (3.12).
- Unsupervised setting: SVD-based technique in Procedure 2 when $k \leq \beta d$ (for arbitrary constant β).

Efficient implementation given samples: In Algorithm 1, a given tensor T is input, and we then perform the updates. However, in many settings (especially machine learning applications), the tensor is not available before hand, and needs to be computed from samples. Computing and storing the tensor can be enormously expensive for high-dimensional problems. Here, we provide a simple observation on how we can manipulate the samples directly to carry out the update procedure in Algorithm 1 as *multi-linear* operations, leading to efficient computational complexity.

Consider the mutiview mixtures model desribed in Section 3.3.1 where the goal is to decompose the empirical moment tensor $\hat{T}$ of the form

$$\widehat{T} := \frac{1}{n} \sum_{l \in [n]} x_1^{(l)} \otimes x_2^{(l)} \otimes x_3^{(l)}, \tag{3.10}$$

where $x_r^{(l)}$ is the l^{th} sample from view $r \in [3]$. Applying the power update (2.6) in Algorithm 1 to $\hat{T}$, we have

$$\tilde{c} := \widehat{T}(\widehat{a}, \widehat{b}, I) = \frac{1}{n} X_3 \big(X_1^\top \widehat{a} * X_2^\top \widehat{b} \big), \tag{3.11}$$

where $*$ corresponds to the *Hadamard* product. Here, $X_r := \big[x_r^{(1)} \ x_r^{(2)} \ \cdots \ x_r^{(n)} \big] \in \mathbb{R}^{d \times n}$. Thus, the update can be computed efficiently using simple matrix and vector operations. It is easy to see that the above update in (3.11) is easily parallelizable, and especially, the different initializations can be parallelized, making the algorithm scalable for large problems.

We now provide some basic assumptions incorporated throughout the learning results, and state the organization of learning guarantees which are proposed in subsequent sections.

Basic assumptions

Here, we review some of the assumptions and settings assumed throughout the learning results provided in next sections. Consider tensor decomposition form in (1.5). Let $A := [a_1 \ a_2 \ \cdots \ a_k] \in \mathbb{R}^{d \times k}$ denote the *factor matrix*. Similar factor matrices are defined as B and C in the asymmetric cases, e.g., multiview linear mixtures model. For simplicity and without loss of generality, we assume that the columns of factor matrices have unit ℓ_2 norm, since we can always rescale them, and adjust the weights appropriately. Also, for simplicity we assume $a_i, b_i, c_i \in \mathbb{R}^d, i \in [k]$, are uniformly i.i.d. drawn from the unit d-dimensional sphere $\mathcal{S}^{d-1}$ (see Remark 4 for more details).

In this chapter, we focus on learning in the challenging overcomplete regime where the number of components/mixtures is larger than observed dimension. Precisely, we assume $k \geq \Omega(d)$. Note that the results can be easily adapted to the highly undercomplete regime when $k \leq o(d)$.

Learning results organization

In Section 3.3, we described how learning different latent variable models can be formulated as a tensor decomposition problem by performing appropriate modifications on the observed moments. For those LVMs, the tensor concentration bounds are provided in Section 3.4. We also proposed the tensor decomposition algorithm in Section 2.3 which is robust to noise. Employing all these techniques and results, we finally provide learning results for different latent variable models including multiview linear mixtures, ICA and sparse ICA in the subsequent sections. We consider two settings, viz., semi-supervised setting, where a small amount of label information is available, and unsupervised setting where such information is not available. In the former setting, we can handle overcomplete mixtures with number of components $k = o(d^{p/2})$, where d is the observed dimension and p is the order of observed moment. In the latter case, our analysis only works when $k \leq \beta d$ for any constant β. See the following two sections for learning guarantees.

3.6 Learning Multiview Linear Mixtures Model

In this section, we provide the semi-supervised and unsupervised learning results for the multiview linear mixtures model described in Section 3.3.1.

3.6.1 Semi-supervised Learning

In the semi-supervised setting, label information is exploited to build good initialization vectors for the tensor decomposition algorithm as follows. Let $x_{1,j}^{(l)}, x_{2,j}^{(l)}, x_{3,j}^{(l)} \in \mathbb{R}^d, j \in [k], l \in [m_j]$, denote $m = \sum_{j\in[k]} m_j$ labeled samples, where the samples with subscript j have label j, i.e., they are generated from hidden state $h = j$. Then, given conditional mean model in (3.1), we can compute the empirical estimate of mixture components as

$$\widehat{a}_j := \frac{1}{m_j} \sum_{l\in[m_j]} x_{1,j}^{(l)}, \quad \widehat{b}_j := \frac{1}{m_j} \sum_{l\in[m_j]} x_{2,j}^{(l)}, \quad \widehat{c}_j := \frac{1}{m_j} \sum_{l\in[m_j]} x_{3,j}^{(l)}, \quad \text{for any } j \in [k]. \tag{3.12}$$

Given n unlabeled samples, let

$$\epsilon_R := \begin{cases} \tilde{O}\left(k\sqrt{d}/\sqrt{n}\right) + \tilde{O}\left(\sqrt{k}/d\right), & \zeta^2 = \Theta(1), \\ \tilde{O}\left(\sqrt{k/n}\right) + \tilde{O}\left(\sqrt{k}/d\right), & \zeta^2 = \Theta\left(\frac{1}{d}\right), \end{cases} \tag{3.13}$$

denote the recovery error. We first provide the settings of Algorithm 1 which include input tensor T, number of iterations N and the initialization setting.

Settings of Algorithm 1 in Theorem 3.7:

- Given n unlabeled samples $x_1^{(i)}, x_2^{(i)}, x_3^{(i)} \in \mathbb{R}^d, i \in [n]$, consider the empirical estimate of 3rd order moment in (3.2) as the input to Algorithm 1.
- Number of iterations: $N = \Theta\left(\log\left(1/\epsilon_R\right)\right)$.
- Initialization: Exploit the empirical estimates in (3.12) as initialization vectors.

Conditions for Theorem 3.7:

- Rank condition: $\Omega(d) \leq k \leq o(d^{3/2})$.
- The columns of factor matrices are uniformly i.i.d. drawn from unit d-dimensional sphere $\mathcal{S}^{d-1}$ (see Remark 4 for more discussion).
- Suppose the distribution of observed variables given hidden state is sub-Gaussian, and the number of labeled samples with label j, denoted by m_j, satisfies [1]

$$m_j \geq \tilde{\Omega}\left(\zeta^2 d\right), \ j \in [k]. \tag{3.14}$$

- Given n unlabeled samples, noise matrices E_A, E_B and E_C satisfy the RIP condition in (RIP) which is satisfied with high probability for many random models (see Remark 3 for details on

[1] In model $\mathcal{S}$, the columns of factor matrices are unit vectors, and therefore, the most reasonable regime of error is when the expected norm of error vector is constant, i.e., $\mathbb{E}\left[\|\zeta\sqrt{d}\varepsilon\|^2\right] = \zeta^2 d \leq O(1)$. But, note that the label complexity holds even if $\zeta^2 d \geq \omega(1)$.

RIP condition). The number of samples n satisfies

$$n \geq \begin{cases} \tilde{\Omega}\left(k^2 d\right), & \zeta^2 = \Theta(1), \\ \tilde{\Omega}\left(k\right), & \zeta^2 = \Theta\left(\frac{1}{d}\right), \end{cases} \tag{3.15}$$

where ζ^2 is the variance of each entry of observation vectors.

Theorem 3.7 (Semi-supervised learning of multiview mixtures model)**.** *Assume the conditions and settings mentioned above hold. Then, the algorithm outputs estimates* $\widehat{A} := [\widehat{a}_1 \cdots \widehat{a}_k] \in \mathbb{R}^{d \times k}$ *and* $\widehat{w} := [\widehat{w}_1 \cdots \widehat{w}_k]^\top \in \mathbb{R}^k$*, satisfying w.h.p.*

$$\left\| \widehat{A} - A \right\|_F \leq \tilde{O}\left(\frac{k}{\sqrt{n}}\right), \quad \|\widehat{w} - w\| \leq \tilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{3.16}$$

Similar error bounds hold for other factor matrices B *and* C*.*

Thus, we provide efficient learning guarantees for overcomplete multiview mixtures in the semi-supervised setting given small number of labeled samples. It is also worth mentioning that there is no dependence on the condition numbers of moment matrices in the sample complexity result.

See Appendix C.1 for the proof.

Column-wise error bounds: In Section 2.1.1, we explain that the algorithm analysis also provides column-wise error bounds with the expense of introducing an additional approximation error. More precisely, we provide stronger guarantees on the column-wise errors as

$$\|\widehat{a}_j - a_j\| \leq \tilde{O}\left(\sqrt{k/n}\right) + \tilde{O}\left(\sqrt{k}/d\right), \quad j \in [k], \tag{3.17}$$

where a $\sqrt{k}$ factor is removed in the first term of bound comparing with the bound in (3.16), but an additional approximation error $\tilde{O}\big(\sqrt{k}/d\big)$ is introduced. See Lemma 2.2 and the corresponding discussions for exact description.

Remark 4 (Random assumption). *In the above learning result, we assume that the mixture components are uniformly i.i.d. drawn from unit* d*-dimensional sphere* $\mathcal{S}^{d-1}$*. This assumption is provided for simplicity, while the original conditions for the recovery guarantees are deterministic (provided*

in Appendix A.2). We show that random matrices satisfy these deterministic assumptions with high probability. Notice the random assumption is reasonable for continuous models including the multiview mixtures model described here. But, it is not appropriate for discrete models where the non-negativity assumptions on the entries of factor matrices are required.

Remark 5 (Minimax sample complexity). Note that the number of labeled samples required is much smaller than the number of unlabeled samples, i.e., $\sum_{j\in[k]} m_j \ll n$. Thus, we provide efficient learning guarantees for overcomplete multiview Gaussian mixtures in the semi-supervised setting under a small number of labeled samples. Furthermore, in the low noise regime $\zeta^2 = \Theta\left(\frac{1}{d}\right)$, the sample complexity bounds for unlabeled samples is $\tilde{\Omega}(k)$, which is the *minimax* bound up to polylog factors.

Remark 6 (Different noise regime). For brevity, both semi-supervised and unsupervised learning results for multiview linear mixtures model in this section are provided in low noise $\zeta^2 = \Theta(1/d)$ and high noise $\zeta^2 = \Theta(1)$ regimes. But, notice that the result for general regime of noise (all different magnitudes of ζ) can be provided according to the general tensor concentration bound proposed in Theorem 3.4.

Remark 7 (Bounded $2 \to 3$ norm assumption). Notice that the bounded $2 \to 3$ norm assumption in tensor concentration bound in Theorem 3.4 is a weaker condition than assuming incoherence property for learning result in Theorem 3.7 which is needed for the algorithm guarantees. Furthermore, it is discussed in Anandkumar et al. [19] that under the assumptions $k \leq o(d^{3/2})$ and uniform draws of columns of A, B and C from unit sphere, the bound on $2 \to 3$ norm is satisfied.

Remark 8 (Spherical Gaussian mixtures). Similar learning results as in Theorem 3.7 hold for the spherical Gaussian mixtures. It is discussed in Section 3.3.2 how learning this model can be reduced to the tensor decomposition problem. Here, the 3rd order empirical (modified) moment $\widehat{M}_3$ in (3.4) is considered as the input of Algorithm 1 with symmetric updates. Thus, we show minimax unlabeled sample complexity for semi-supervised learning of overcomplete spherical Gaussian mixtures.

3.6.2 Unsupervised Learning

In the unsupervised setting, there is no label information available to build the initialization vectors. Here, the initialization is performed by doing rank-1 SVD on random slices of the moment tensor proposed in Procedure 2. The conditions and settings for unsupervised learning are stated as follows where comparing to the semi-supervised learning, the initialization setting, rank and sample complexity conditions are changed.

Settings of Algorithm 1 in Theorem 3.8:

- Given n unlabeled samples $x_1^{(i)}, x_2^{(i)}, x_3^{(i)} \in \mathbb{R}^d, i \in [n]$, consider the empirical estimate of 3rd order moment in (3.2) as the input to Algorithm 1.
- Number of iterations: $N = \Theta\left(\log\left(1/\epsilon_R\right)\right)$.
- The initialization in each run of Algorithm 1 is performed by SVD-based technique proposed in Procedure 2, with the number of initializations as

$$L \geq k^{\Omega\left(k^2/d^2\right)}.$$

Conditions for Theorem 3.8:

- Rank condition: $k = \Theta(d)$.
- The columns of factor matrices are uniformly i.i.d. drawn from unit d-dimensional sphere $\mathcal{S}^{d-1}$.
- The number of samples n satisfies

$$n \geq \begin{cases} \tilde{\Omega}\left(k^4\right), & \zeta^2 = \Theta(1), \\ \tilde{\Omega}\left(k^2\right), & \zeta^2 = \Theta\left(\frac{1}{d}\right). \end{cases}$$

Theorem 3.8 (Unsupervised learning of multiview mixtures model)**.** *Assume the conditions and settings mentioned above hold. Suppose the number of unlabeled samples n satisfies $n \geq \tilde{\Omega}\left(kd\right)$. If rank condition $k = \Theta(d)$ holds, then the same guarantees as in Theorem 3.7 are satisfied.*

See Appendix C.1 for the proof.

Remark 9 (Comparison with "whitening + moment-based" techniques in the undercomplete setting when $k \approx d$). Here, we discuss how our approach makes a huge improvement on sample complexity for learning multiview linear mixtures model and spherical Gaussian mixtures with the additional *incoherence* property we assume.

Multiview linear mixtures model: We compare with the previous result by Song et al. [147], which employs whitening procedure followed by tensor power updates in the undercomplete setting. When $k \approx d$, the sample complexity in [147] is scaled as $n \geq \tilde{\Omega}(k^{6.5})$. In comparison, the sample complexity for our method scales as $\tilde{\Omega}(k^2)$, which is far better. This is especially relevant in the high dimensional regime, where k and d are large, and our analysis shows lower sample complexity under incoherent factors.

Spherical Gaussian mixtures: As mentioned in Remark 8, the above unsupervised learning result can be also adapted for learning mixture of spherical Gaussians. An algorithm for learning mixture of spherical Gaussians in the undercomplete setting is also provided in [94], which is a moment-based technique combined with a whitening step. When $k = d$, the sample complexity in [94] scales as $n \geq \tilde{\Omega}(k^3)$. But, our tight tensor concentration analysis leads to the better sample complexity of $n \geq \tilde{\Omega}(k^2)$. Note that this comparison is in the low noise regime $\zeta^2 = \Theta\big(\frac{1}{d}\big)$.

Remark 10 (Extension to $k \leq o(d^{1.5})$). *We also argue that the SVD initialization can be slightly modified, and under some regime of noise we can extend the above unsupervised learning result to the highly overcomplete regime $k \leq o(d^{1.5})$. Suppose the expected norm of noise is constant (low noise regime), and the noise vectors are incoherent with the true mean components (which is satisfied for random mean components). Then if the SVD initialization is performed using samples*[2] *$x_3^{(i)}$, then the same guarantees as in Theorem 3.8 hold under highly overcomplete regime $k \leq o(d^{1.5})$.*

[2]The SVD of $T\big(I, I, x_3^{(i)}\big)$ is computed.

3.7 Learning Multiview Mixture Model Under Random Means

Assuming random rank-1 components, we provided stronger convergence guarantees for the tensor power iteration in Chapter 2; see Theorem 2.3. Along this result we provide the application to learning multiview mixtures model in Theorem 3.9.

Consider the same multiview mixture model as proposed in Section 3.3.1 with the difference that the noise is not necessarily Gaussian. The variables (views) $x_l \in \mathbb{R}^d$ are related to the hidden state through *factor matrix* $A \in \mathbb{R}^{d\times k}$ such that

$$x_l = Ah + \eta_l, \quad l \in [p],$$

where hidden state h is represented as j-th basis vector e_j if the hidden variable takes j-th state. $\eta_l \in \mathbb{R}^d$ denote zero-mean noise vectors which are independent of each other and the hidden state h.

For the learning algorithm, consider the same tensor decomposition algorithm proposed in Section 2.3 with the modification that the algorithm is initialized with random samples of observed variables. The algorithm is initialized by all n samples, and the final output is computed by the clustering algorithm. More details are provided in the next section.

3.7.1 Learning guarantees

We assume a Gaussian prior on the mean vectors, i.e., the vectors $a_j \sim \mathcal{N}(0, I_d/d)$, $j \in [k]$ are i.i.d. drawn from a standard multivariate Gaussian distribution with unit expected square norm. Note that in the high dimension (growing d), this assumption is the same as uniformly drawing from unit sphere since the norm of vector concentrates in the high dimension and there is no need for normalization. Even though we impose a prior distribution, we do not use a MAP estimator, since the corresponding optimization is NP-hard. Instead, we learn the model parameters through decomposition of the third order moments through tensor power iterations. The assumption of a Gaussian prior is standard in machine learning applications. We impose it here for tractable analysis

of power iteration dynamics. Such Gaussian assumptions have been used before for analysis of other iterative methods such as approximate message passing algorithms, and there are evidences that similar results hold for more general distributions; see [39] and references there.

As explained in the previous sections, we use tensor power method to learn the components a_j's, and the method is initialized with observed samples x_i. Intuitively, this initialization is useful since $x_i = Ah + \eta_i$ is a perturbed version of desired parameter a_j (when $h = e_j$). Thus, we present the result in terms of the signal-to-noise (SNR) ratio which is the expected norm of signal a_j (which is one here) divided by the expected norm of noise η_i, i.e., the SNR in the i-th sample $x_i = a_j + \eta_i$ (assumed $h = e_j$) is defined as $\text{SNR} := \mathbb{E}[\|a_j\|]/\mathbb{E}[\|\eta_i\|]$. This specifies how much noise the initialization vector x_i can tolerate in order to ensure the convergence of tensor power iteration to a desired local optimum. We now propose the conditions required for recovery guarantees, and state a brief explanation of them.

Conditions for Theorems 3.9 and 3.10:

- Rank condition: $k \leq o(d^{1.5})$.
- The columns of A are uniformly i.i.d. drawn from unit d-dimensional sphere.
- The noise vectors $\eta_l, l \in [3]$, are independent of matrix A and each other. In addition, the signal-to-noise ratio (SNR) is w.h.p. bounded as

  $$\text{SNR} \geq \Omega\left(\frac{\sqrt{\max\{k,d\}}}{d^{1-\beta}}\right),$$

  for some $\beta \geq (\log d)^{-c}$ for universal constant $c > 0$.

The rank condition bounds the level of overcompleteness for which the recovery guarantees are satisfied. The random assumption on the columns of A are crucial for analyzing the dynamics of tensor power iteration. We use it to argue there exists enough randomness left in the components after conditioning on the previous iterations; see Section 2.6.1 for the details. The bound on the SNR is required to make sure the given sample used for initialization is close enough to the corresponding mean vector. This ensures that the initial vector is inside the basin-of-attraction of

the corresponding component, and hence, the convergence to the mean vector can be guaranteed. Under these assumptions we have.

Theorem 3.9 (Learning multiview mixture model given exact tensor: closeness to single columuns)**.** *Consider a multiview mixture model (or a spherical Gaussian mixture) in the above setting with k components in d dimensions. If the above conditions hold, then the tensor power iteration converges to a vector close to one of the true mean vectors a_j's (having constant correlation).*

In particular, for mildly overcomplete models, where $k = \alpha d$ for some constant $\alpha > 1$, the signal-to-noise ratio (SNR) is as low as $\Omega(d^{-1/2+\epsilon})$, for any $\epsilon > 0$. Thus, we can learn mixture models with a high level of noise. In general, we establish how the required noise level scales with the number of hidden components k, as long as $k = o(d^{1.5})$.

The above theorem states convergence to desired local optima which are close to true components a_j's. In Theorem 3.10, we show that we can sharpen the above result, by jointly iterating over the recovered vectors, and consistently recover the components a_j's. This result also uses the analysis in the previous section.

Theorem 3.10 (Learning multiview mixture model given exact tensor: recovering the whole factor matrix)**.** *Assume the above conditions hold. The initialization of power iteration is performed by samples of x_1 in multiview mixture model. Suppose the tensor power iterations is at least initialized once for each $a_j, j \in [k]$ such that $x_1 = a_j + \eta_1$.*[3] *Then by using the exact 3rd order moment tensor in* (3.2) *as input, the tensor decomposition algorithm outputs an estimate $\widehat{A}$ satisfying w.h.p. (over the randomness of the components a_j's)*

$$\left\| \widehat{A} - A \right\|_F \leq \epsilon,$$

where the number of iterations of the algorithm is $N = \Theta\left(\log\left(\frac{1}{\epsilon}\right) + \log\log d\right)$.

The above theorems assume the exact third order tensor is given to the algorithm. We provide the results given empirical tensor in Section 3.7.1.1.

[3]Note that this happens for component j with high probability when the number of initializations is proportional to inverse prior probability corresponding to that mixture.

3.7.1.1 Sample complexity analysis

In the previous section, we assumed the exact third order tensor in (3.2) is given to the tensor decomposition Algorithm 1. We now estimate the tensor given n samples $x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, i \in [n]$, as

$$\widehat{T} = \frac{1}{n} \sum_{i \in [n]} x_1^{(i)} \otimes x_2^{(i)} \otimes x_3^{(i)}. \tag{3.18}$$

For the multiview mixture model introduced in Section 3.3.1, let the noise vector η_l be spherical, and ζ^2 denote the variance of each entry of noise vector. We now provide the following recovery guarantees.

Additional conditions for Theorem 3.11:

- Let $E_1 := [\eta_1^{(1)}, \eta_1^{(2)}, \ldots, \eta_1^{(n)}] \in \mathbb{R}^{d \times n}$, where $\eta_1^{(i)} \in \mathbb{R}^d$ is the i-th sample of noise vector η_1. These noise matrices satisfy the following *RIP property* which is adapted from Candes and Tao [47]. Matrix $E_1 \in \mathbb{R}^{d \times n}$ satisfies a weak RIP condition such that for any subset of $O\left(\frac{d}{\log^2 d}\right)$ number of columns, the spectral norm of E_1 restricted to those columns is bounded by 2. The same condition is satisfied for similarly defined noise matrices E_2 and E_3.
- The number of samples n satisfies lower bound such that

$$\zeta\left(\frac{\sqrt{d}}{n} + \sqrt{\lambda_{\max}\frac{d}{n}}\right) + \zeta^2\left(\frac{d}{n} + \sqrt{\lambda_{\max}\frac{d^{1.5}}{n}}\right) + \zeta^3\left(\frac{d^{1.5}}{n} + \sqrt{\frac{d}{n}}\right) \leq \min\left\{\epsilon\frac{\sqrt{k}}{d}, \tilde{O}(\lambda_{\min})\right\}, \tag{3.19}$$

  where $\epsilon < o\left(\sqrt{k}/d\right)$.

Theorem 3.11 (Learning multiview mixture model given empirical tensor)**.** *Consider the empirical tensor in* (3.18) *as the input to tensor decomposition Algorithm 1. Suppose the above additional conditions are also satisfied. Then, the same guarantees as in Theorem 3.9 hold. In addition, the same guarantees as in Theorem 3.10 also hold with the recovery bound changed as*

$$\left\|\widehat{A} - A\right\|_F \leq \tilde{O}\left(\frac{\sqrt{k} \cdot \|E\|}{\lambda_{\min}}\right),$$

where E denotes the perturbation tensor originated from empirical estimation in (3.18)*, and its spectral norm $\|E\|$ is bounded by the LHS of* (3.19)*.*

Proof: The above sample complexity result is proved by using the tensor concentration bound in Theorem 1 of Anandkumar et al. [21] applied to our noisy analysis of tensor power dynamics in Theorem 2.3; see Equation (2.4). The additional bound on sample complexity and final recovery error on $\left\|\widehat{A} - A\right\|_F$ is also from Theorem 1 of Anandkumar et al. [16]. □

3.8 Learning Independent Component Analysis (ICA) and Sparse ICA

In this section, we propose the semi-supervised and unsupervised learning results for the ICA and sparse ICA models. By semi-supervised setting in ICA, we mean some prior information is available which provides good initializations for the components. Recall the standard ICA model [59], where *independent* source signals are linearly mixed to generate the observations. Let $h \in \mathbb{R}^k$ be a random latent signal where its coordinates are independent, and $A \in \mathbb{R}^{d\times k}$ be the mixing matrix. Then, the observed vector is

$$x = Ah \in \mathbb{R}^d.$$

For simplicity, we limit to noiseless setting. This is the standard setting, and is already challenging because samples in ICA are mixtures of many components, unlike the mixture models. It is discussed in Section 3.3.3 how estimating the parameters of ICA model can be formulated as a tensor decomposition problem where a modified version of 4th order observed moment (denoted by M_4) is characterized in a tensor decomposition form; see Lemma 3.2.

We now provide the learning results for the *sparse* ICA problem which is more general. This is the ICA setting with the assumption that hidden vector $h \in \mathbb{R}^k$ can be sparse with i.i.d. Bernoulli-subgaussian random entries. Assume the probability of each Bernoulli variable being 1 is s/k. Note

that (dense) ICA is special case when $s = k$. For the sparse ICA model, we also assume that mixing matrix A satisfies the RIP property (see condition (RIP) in Section 3.3.1).

Settings of Algorithm in Theorem 3.12: Given n samples $x^i = Ah^i, i \in [n]$, consider the empirical estimate of 4th order (modified) moment M_4 (see (3.5) in the Appendix) as the input to the algorithm with symmetric 4th order updates; see Appendix 2.4.1 for higher order extension of the algorithm. Let the number of iterations $N = \tilde{\Theta}\left(\log\left(1/\tilde{\epsilon}_R\right)\right)$, where $\tilde{\epsilon}_R := \min\left\{k^2/\min\left\{n, \sqrt{d^3 n}\right\}, \sqrt{k}/d^{1.5}\right\}$. The initialization is performed differently in different learning settings. In the *semi-supervised* setting, it is assumed that for any $j \in [k]$, an approximation of a_j denoted by $\widehat{a}_j^{(0)}$ is given satisfying $\|\widehat{a}_j^{(0)} - a_j\| \leq \alpha$ for some constant $\alpha < 1$. In the *unsupervised* setting, the initialization is performed by 4-th order generalization [4] of SVD-based technique in Procedure 2, with the number of initializations as $L \geq k^{\Omega(k^2/d^2)}$.

Theorem 3.12 (Semi-supervised and unsupervised learning of (sparse) ICA)**.** *Assume the Algorithm settings mentioned above hold. In the* semi-supervised *setting, suppose*

$$n \geq \begin{cases} \tilde{\Omega}(sk), & sk \leq O(d^3)/\operatorname{polylog}(d), \\ \tilde{\Omega}\left(s^2k^2/d^3\right), & \text{o.w.}, \end{cases}$$

and rank condition $\Omega(d) \leq k \leq o(d^2)$ hold. In the unsupervised *setting, suppose $n \geq \tilde{\Omega}\left(k^2 s\right)$, and rank condition $\Omega(d) = \Theta(d)$ hold. Then the algorithm outputs estimates $\widehat{A}$ and $\widehat{w}$, satisfying w.h.p.*

$$\max\left\{\left\|\widehat{A} - A\right\|_F, \|\widehat{w} - w\|\right\} \leq \tilde{O}\left(\frac{s \cdot k^{1.5}}{\min\left\{n, \sqrt{d^3 n}\right\}}\right).$$

In one extreme when $s = \Theta(k)$, it is akin to learning the "dense" ICA model.[5] On the other extreme when s is a constant, it is akin to learning multiview models. Thus, the sparse coding model bridges the range of models between multiview mixtures model and ICA.

[4] In the 4th order case, the SVD is performed on $T(I, I, \theta, \theta) \in \mathbb{R}^{d \times d}$ for some random vector θ.

[5] The result for learning ICA is a special case when $s = k$. Note that since we provide a different proof for the ICA model, it does not need the RIP condition on dictionary matrix.

Similar to the multiview mixture model, we can also provide column-wise recovery guarantees with introducing additional approximation error $\tilde{O}\big(\sqrt{k}/d^{1.5}\big)$. Note that this error is different from multiview mixture since we exploit different tensor orders in the two models.

Comparison with previous approaches: The dictionary learning problem is also studied in Arora et al. [27], Agarwal et al. [3], Barak et al. [34]. Arora et al. [27], Agarwal et al. [3] provide clustering based approaches for approximately learning incoherent dictionaries and then refining them through alternating minimization to obtain exact recovery of both the dictionary and the coefficients. They can handle sparsity level up to $O(\sqrt{d})$ (per sample) and the size of the dictionary k can be arbitrary. Barak et al. [34] use the sum of squares framework and can handle the sparsity level up to (small enough) constant times k, but with the expense of computational complexity which scales as $k^{O(\log k)}$, and the size of the dictionary $k = O(d)$. In addition, when the sparsity level is smaller as $k^{1-\delta}$ for some $0 < \delta < 1$, their algorithm runs in polynomial time $k^{O(1/\delta)}$. They can also go to higher level of overcompleteness with the expense of reducing sparsity level. They do not need the assumptions that the dictionary is incoherent or that the coefficients are independent. They only have approximate recovery and note that exact recovery is impossible (from an identifiability standpoint) unless further assumptions are imposed. In contrast, we have a polynomial time method for incoherent dictionaries and independent coefficients which can handle arbitrary sparsity level, and provides approximate recovery. Moreover, we can handle larger dictionary sizes k at the expense of more computation.

Below, we show how we can extend our analysis to dependent sparsity setting, but with worse performance guarantees.

Extension to dependent sparsity

In this section, we consider the noiseless sparse coding model $x = Ah$, but with no independence assumption on the latent entries h_i's. The analysis can be extended to noisy case.

We assume the following moment conditions on h in the dependent sparsity model. Note that these assumptions are comparable with the moment assumptions in Barak et al. [34].

$$\begin{aligned}\mathbb{E}\big[h_i^4\big] = \mathbb{E}\big[h_i^2\big] &= \beta s/k,\\ \mathbb{E}\big[h_i^2 h_j^2\big] &\le \tau, \quad i \neq j,\\ \mathbb{E}\big[h_i^3 h_j\big] &= 0, \quad i \neq j,\end{aligned}$$

with parameters s and τ, where s is the expected number of nonzero entries in h, and β is a universal constant. The first condition represents the normalization factor which depends on the sparsity level. The second condition limits the sparsity level and the amount of correlation between different entries of vector h. To provide more intuition about these parameters, assume that the entries of h are distributed as Bernoulli-Gaussian random variables with each entry being nonzero with probability s/k. Then, we have $\tau = \rho p + (1-\rho)p^2$, where ρ is the correlation coefficient between h_i^2 and h_j^2 for $i \neq j$.

Theorem 3.13 (Noiseless sparse coding with dependent sparsity)**.** *Consider the described dictionary learning model $x = Ah$ where the moments of random vector h satisfy the conditions stated before the theorem. Let the noiseless 4th order observed moment $\mathbb{E}\big[x^{\otimes 4}\big]$ be the input to Algorithm 1 with symmetric 4th order updates. Let the initialization in each run of Algorithm 1 is performed by 4th order generalization of the SVD-based technique proposed in Procedure 2. Let $\tilde{\epsilon}_R := \tilde{O}(\tau k/s) + \tilde{O}\big(\sqrt{k}/d^{3/2}\big)$, and suppose*

$$k = \Theta(d), \quad N = \Theta\left(\log\left(1/\tilde{\epsilon}_R\right)\right), \quad L \ge k^{\Omega\left(k^2/d^2\right)}.$$

In addition, assume that the columns of dictionary A are uniformly i.i.d. drawn from unit d-dimensional sphere $\mathcal{S}^{d-1}$. If

$$\tau \le \tilde{O}\left(\frac{s/k}{d}\right),$$

then whp

$$\text{dist}\,(\hat{a}_j, a_j) \leq \tilde{\epsilon}_R, \quad j \in [k].$$

See Appendix C.1 for the proof.

Comparing with the dictionary learning result by Barak et al. [34], their algorithm is based on sum-of-squares techniques, and do not require any incoherence assumptions on the dictionary atoms. They can also handle higher levels of sparsity and correlation. On the other hand, they have a quasi-polynomial algorithm in the regime of high sparsity (small enough constant times k), while our algorithm is very simple and efficient.

The above analysis is in the noiseless regime, and the generalization to noisy case can be investigated as a future work which involves the sample complexity analysis in the dependent sparsity case.

3.9 Experiments

In this Section, we run the algorithm for learning multiview Gaussian mixtures model. We consider model $\mathcal{S}$ described in Section 3.3.1. The mixture components are uniformly i.i.d. drawn from d-dimensional sphere $\mathcal{S}^{d-1}$. We assume low-noise regime such that $\zeta\sqrt{d} = 0.1$. In addition, let [6] $w_j = \Pr[h = j] = \frac{1}{k}, j \in [k]$. We consider $d = 100$ and $k = \{10, 20, 50, 100, 200, 500\}$. In order to see the effect of number of components k, we fix the number of samples $n = 1000$.

Notice that the empirical tensor $\widehat{T}$ in (3.10) is not explicitly computed, and the tensor power updates in the algorithm are computed through the multilinear form stated in (3.11). This leads to efficient computational complexity. See Section 2.3 for detailed discussion.

For each initialization $\tau \in [L]$, an alternative option of running the algorithm with a fixed number of iterations N is to stop the iterations based on some stopping criteria. In this experiment, we

[6] In order to see the algorithm performance more easily, we generate n samples such that each mixture component is exactly appeared in $\frac{n}{k}$ observations. Note that this is basically imposing equal number of different mixture components in the observations.

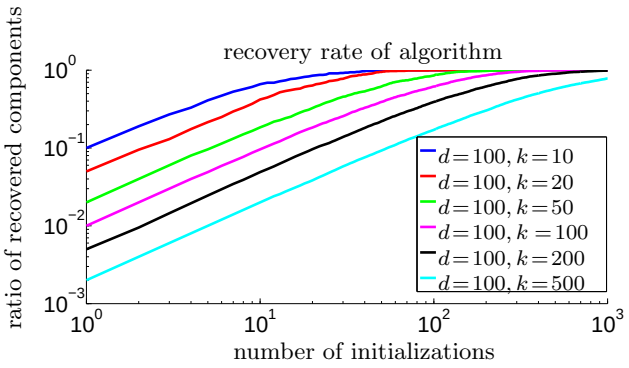


Figure 3.3: Ratio of recovered components vs. the number of initializations. The figure is an average over 10 random runs.

stop the iterations when the improvement in subsequent steps is small as

$$\max\left(\left\|\widehat{a}_\tau^{(t)}-\widehat{a}_\tau^{(t-1)}\right\|^2,\left\|\widehat{b}_\tau^{(t)}-\widehat{b}_\tau^{(t-1)}\right\|^2,\left\|\widehat{c}_\tau^{(t)}-\widehat{c}_\tau^{(t-1)}\right\|^2\right)\leq t_{\mathrm{S}},$$

where t_{S} is the stopping threshold. According to the error bound provided in Theorem 3.7, we let

$$t_{\mathrm{S}}:=t_1(\log d)^2\sqrt{\frac{k}{n}}+t_2(\log d)^2\frac{\sqrt{k}}{d}, \tag{3.20}$$

for some constants $t_1,t_2>0$. Here, we set $t_1=1e-08$, and $t_2=1e-07$.

A random initialization approach is used where $\widehat{a}^{(0)}$ and $\widehat{b}^{(0)}$ are uniformly i.i.d. drawn from sphere $\mathcal{S}^{d-1}$. Initialization vector $\widehat{c}^{(0)}$ is generated through update formula in (2.6). Figure 3.3 depicts the ratio of recovered components vs. the number of initializations. We observe that the algorithm is capable of recovering mixture components even in the overcomplete regime $k\geq d$. As suggested in the experimental results of Anandkumar et al. [19], we also observe that random initialization works efficiently in the experiments, while the theoretical results for random initialization appear to be highly pessimistic. This suggests additional room for improving the theoretical guarantees under random initialization.

Table 3.1 provides the average square error of the estimates, the average weight error and the average number of iterations for different values of k. The averages are over different initializations

Table 3.1: Results for learning a multi-view mixture model. $d = 100$, $n = 1000$, $\zeta\sqrt{d} = 0.1$.

k	avg. square error	avg. weight error	avg. # of iterations	avg. square error $/k$	avg. weight error $/k$
10	1.24e-03	1.73e-05	9.81	1.24e-04	1.73e-06
20	2.94e-03	5.28e-05	10.98	1.41e-04	2.64e-06
50	7.21e-03	1.84e-04	12.74	1.44e-04	3.69e-06
100	1.47e-02	5.36e-04	14.86	1.47e-04	5.36e-06
200	3.03e-02	1.85e-03	18.34	1.51e-04	9.23e-06
500	8.26e-02	1.23e-02	30.02	1.65e-04	2.45e-05

and random runs. The square error is computed as

$$\frac{1}{3}\left[\|a_j - \widehat{a}\|^2 + \left\|b_j - \widehat{b}\right\|^2 + \|c_j - \widehat{c}\|^2\right],$$

for the corresponding recovered column j. The weight error is computed as square relative error $|\widehat{w} - w_j|^2/w_j^2$. The number of iterations performed before stopping the algorithm is mentioned in the fourth column. We observe that we can still get good error bounds even for overcomplete models with $d = 100$ and $k = 500$.

In the last two columns, the normalized values of errors are provided. The normalization is done by the number of mixtures k. Here, we observe that the normalized values (specially for the square error) are very close for different k. This complies with the theoretical error bound in (3.17) which claims that the square recovery error is bounded as $\tilde{O}(k)$ when d and n are fixed as here.

Chapter 4

Training Neural Networks Using Tensor Methods

Training neural networks is a challenging non-convex optimization problem, and backpropagation or gradient descent can get stuck in spurious local optima. In this chapter, we propose a novel algorithm based on tensor decomposition for guaranteed training of two-layer neural networks. We provide risk bounds for our proposed method, with a polynomial sample complexity in the relevant parameters, such as input dimension and number of neurons. While learning arbitrary target functions is NP-hard, we provide transparent conditions on the function and the input for learnability. Our training method is based on tensor decomposition, which provably converges to the global optimum, under a set of mild non-degeneracy conditions; the details are provided in Chapter 2. It consists of simple embarrassingly parallel linear and multi-linear operations, and is competitive with standard stochastic gradient descent (SGD), in terms of computational complexity. Thus, we propose a computationally efficient method with guaranteed risk bounds for training neural networks with one hidden layer.

Note that the analysis in this chapter has a fundamental difference with the learning results provided in Chapter 3 such that here the problem of training neural networks is in supervised setting, while learning latent variable models in Chapter 3 are unsupervised or semi-supervised.

Neural networks have revolutionized performance across multiple domains such as computer vision and speech recognition. They are flexible models trained to approximate any arbitrary target function, e.g., the label function for classification tasks. They are composed of multiple layers of *neurons* or *activating functions*, which are applied recursively on the input data, in order to predict the output. While neural networks have been extensively employed in practice, a complete theoretical understanding is currently lacking.

Training a neural network can be framed as an optimization problem, where the network parameters are chosen to minimize a given loss function, e.g., the *quadratic loss* function over the error in predicting the output. The performance of training algorithms is typically measured through the notion of *risk*, which is the expected loss function over unseen test data. A natural question to ask is the hardness of training a neural network with a bounded risk. The findings are mostly negative [137, 145, 44, 37, 113]. Training even a simple network is NP-hard, e.g., a network with a single neuron [145].

The computational hardness of training is due to the non-convexity of the loss function. In general, the loss function has many *critical points*, which include *spurious local optima* and *saddle points*. In addition, we face *curse of dimensionality*, and the number of critical points grows exponentially with the input dimension for general non-convex problems [67]. Popular local search methods such as gradient descent or *backpropagation* can get stuck in bad local optima and experience arbitrarily slow convergence. Explicit examples of its failure and the presence of bad local optima in even simple separable settings have been documented before [46, 79, 75]; see Section 4.7.1 for a discussion.

Alternative methods for training neural networks have been mostly limited to specific activation functions (e.g., linear or quadratic), specific target functions (e.g., polynomials) [24], or assume strong assumptions on the input (e.g., Gaussian or product distribution) [24], see related work for details. Thus, up until now, there is no unified framework for training networks with general input, output and activation functions, for which we can provide guaranteed risk bound.

In this chapter, for the first time, we present a guaranteed framework for learning general target functions using neural networks, and simultaneously overcome computational, statistical, and approximation challenges. In other words, our method has a low computational and sample complexity, even as the dimension of the optimization grows, and in addition, can also handle approximation errors, when the target function may not be generated by a given neural network. We prove a guaranteed risk bound for our proposed method. NP-hardness refers to the computational complexity of training worst-case instances. Instead, we provide transparent conditions on the target functions and the inputs for tractable learning.

Our training method is based on the method of moments, which involves decomposing the empirical cross moment between output and some function of input. While pairwise moments are represented using a matrix, higher order moments require tensors, and the learning problem can be formulated as tensor decomposition. A CP (CanDecomp/Parafac) decomposition of a tensor involves finding a succinct sum of rank-one components that best fit the input tensor. Even though it is a non-convex problem, the global optimum of tensor decomposition can be achieved using computationally efficient techniques, under a set of mild non-degeneracy conditions [20, 18, 22, 17, 41]. These methods have been recently employed for learning a wide range of latent variable models [18, 13].

Incorporating tensor methods for training neural networks requires addressing a number of non-trivial questions: What form of moments are informative about network parameters? Earlier works using tensor methods for learning assume a linear relationship between the hidden and observed variables. However, neural networks possess non-linear activation functions. How do we adapt tensor methods for this setting? How do these methods behave in the presence of approximation and sample perturbations? How can we establish risk bounds? We address these questions shortly.

4.1 Summary of Results

The main contributions are: (a) we propose an efficient algorithm for training neural networks, termed as Neural Network-LearnIng using Feature Tensors (NN-LIFT), (b) we demonstrate that the method is embarrassingly parallel and is competitive with standard SGD in terms of computational

complexity, and as a main result, (c) we establish that it has bounded risk, when the number of training samples scales polynomially in relevant parameters such as input dimension and number of neurons.

We analyze training of a two-layer feedforward neural network, where the second layer has a linear activation function. This is the classical neural network considered in a number of works [64, 93, 36], and a natural starting point for the analysis of any learning algorithm. Note that training even this two-layer network is non-convex, and finding a computationally efficient method with guaranteed risk bound has been an open problem up until now.

At a high level, NN-LIFT estimates the weights of the first layer using tensor CP decomposition. It then uses these estimates to learn the bias parameter of first layer using a simple Fourier technique, and finally estimates the parameters of last layer using linear regresion. NN-LIFT consists of simple linear and multi-linear operations [18, 22, 17], Fourier analysis and ridge regression analysis, which are parallelizable to large-scale data sets. The computational complexity is comparable to that of the standard SGD; in fact, the parallel time complexity for both the methods is in the same order, and our method requires more processors than SGD by a multiplicative factor that scales linearly in the input dimension.

Generative vs. discriminative models: Generative models incorporate a joint distribution $p(x, y)$ over both the input x and label y. On the other hand, discriminative models such as neural networks only incorporate the conditional distribution $p(y|x)$. While training neural networks for general input x is NP-hard, **does knowledge about the input distribution $p(x)$ make learning tractable?**

In this work, we assume knowledge of the input density $p(x)$, which can be any continuous differentiable function. Unlike many theoretical works, e.g., [24], we do not limit ourselves to distributions such as product or Gaussian distributions for the input. While unsupervised learning, i.e., estimation of density $p(x)$, is itself a hard problem for general models, in this work, we investigate how $p(x)$ can be exploited to make training of neural networks tractable. The knowledge of $p(x)$ is naturally available in the *experimental design* framework, where the person designing the experiments

has the ability to choose the input distribution. Examples include conducting polling, carrying out drug trials, collecting survey information, and so on.

Utilizing generative models on the input via score functions: We utilize the knowledge about the input density $p(x)$ (up to normalization)[1] to obtain certain (non-linear) transformations of the input, given by the class of score functions. *Score functions* are normalized derivatives of the input pdf; see (4.5). If the input is a vector (the typical case), the first order score function (i.e., the first derivative) is a vector, the second order score is a matrix, and the higher order scores are tensors. In our NN-LIFT method, we first estimate the cross-moments between the output and the input score functions, and then decompose it to rank-1 components.

Risk bounds: Risk bound includes both approximation and estimation errors. The approximation error is the error in fitting the target function to a neural network of given architecture, and the estimation error is the error in estimating the weights of that neural network using the given samples.

We first consider the *realizable setting* where the target function is generated by a two-layer neural network (with hidden layer of neurons consisting of any general sigmoidal activations), and a linear output layer. Note that the approximation error is zero in this setting. Let $A_1 \in \mathbb{R}^{d\times k}$ be the weight matrix of first layer (connecting the input to the neurons) with k denoting the number of neurons and d denoting the input dimension. Suppose these weight vectors are non-degenerate, i.e., the weight matrix A_1 (or its tensorization) is full column rank. We assume continuous input distribution with access to score functions, which are bounded on any set of non-zero measure. We allow for any general sigmoidal activation functions with non-zero third derivatives in expectation, and satisfying Lipschitz property. Let $s_{\min}(\cdot)$ be the minimum singular value operator, and $M_3(x) \in \mathbb{R}^{d\times d^2}$ denote the matricization of input score function tensor $\mathcal{S}_3(x) \in \mathbb{R}^{d\times d\times d}$; see (1.1) and (4.5) for the definitions. For the Gaussian input $x \sim \mathcal{N}(0, I_d)$, we have $\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right] = \tilde{O}\left(d^3\right)$. We have the following learning result in the realizable setting where the target function is generated by a two layer neural network (with one hidden layer).

[1] We do not require the knowledge of the normalizing constant or the partition function, which is $\#P$ hard to compute [157].

Theorem 4.1 (Informal result for realizable setting)**.** *Our method NN-LIFT learns a realizable target function up to error ϵ when the number of samples is lower bounded as*[2]*,*

$$n \geq \tilde{O}\left(\frac{k}{\epsilon^2} \cdot \mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right] \cdot \frac{s_{\max}^2(A_1)}{s_{\min}^6(A_1)}\right).$$

Thus, we can efficiently learn the neural network parameters with polynomial sample complexity using NN-LIFT algorithm. In addition, the method has polynomial computational complexity, and in fact, its parallel time complexity is the same as stochastic gradient descent (SGD) or backpropagation. See Theorem 4.3 for the formal result.

We then extend our results to the *non-realizable setting* where the target function *need not be* generated by a neural network. For our method NN-LIFT to succeed, we require the approximation error to be sufficiently small under the given network architecture. Note that it is not of practical interest to consider functions with large approximation errors, since classification performance in that case is poor [38]. We state the informal version of the result as follows.

We assume the following: the target function $f(x)$ has a continuous Fourier spectrum and is sufficiently smooth, i.e., the parameter C_f (see (4.10) for the definition) is sufficiently small as specified in (4.15). This implies that the approximation error of the target function can be controlled, i.e., there exists a neural network of given size that can fit the target function with bounded approximation error. Let the input x be bounded as $\|x\| \leq r$. Our informal result is as follows. See Theorem 4.5 for the formal result.

Theorem 4.2 (Informal result for non-realizable setting)**.** *The arbitrary target function $f(x)$ is approximated by the neural network $\widehat{f}(x)$ which is learnt using NN-LIFT algorithm such that the risk bound satisfies w.h.p.*

$$\mathbb{E}_x[|f(x) - \widehat{f}(x)|^2] \leq O(r^2 C_f^2) \cdot \left(\frac{1}{\sqrt{k}} + \delta_1\right)^2 + O(\epsilon^2),$$

where k is the number of neurons in the neural network, and δ_τ is defined in (4.13)*.*

[2]Here, only the dominant terms in the sample complexity are noted; see (4.9) for the full details.

In the above bound, we require for the target function $f(x)$ to have bounded first order moment in the Fourier spectrum; see (4.15). As an example, we show that this bound is satisfied for the class of scale and location mixtures of the Gaussian kernel function.

Corollary 4.1 (Learning mixtures of Gaussian kernels)**.** *Let* $f(x) := \int K(\alpha(x+\beta))G(d\alpha, d\beta)$, $\alpha > 0$, $\beta \in \mathbb{R}^d$, *be a location and scale mixture of the Gaussian kernel function* $K(x) = \exp\left(-\frac{\|x\|^2}{2}\right)$, *the input be Gaussian as* $x \sim \mathcal{N}(0, \sigma_x^2 I_d)$, *and the activations be step functions, then, our algorithm trains a neural network with risk bounds as in Theorem 4.2, when*

$$\int |\alpha| \cdot |G|(d\alpha, d\beta) \leq \text{poly}\left(\frac{1}{d}, \frac{1}{k}, \epsilon, \frac{1}{\sigma_x}, \exp\left(-1/\sigma_x^2\right)\right).$$

We observe that when the kernel mixtures correspond to smoother functions (smaller α), the above bound is more likely to be satisfied. This is intuitive since smoother functions have lower amount of high frequency content. Also, notice that the above bound has a dependence on the variance of the Gaussian input σ_x. We obtain the most relaxed bound (r.h.s. of above bound) for middle values of σ_x, i.e., when σ_x is neither too large nor too small. See Appendix D.3.1 for more discussion and the proof of the corollary.

Intuitions behind the conditions for the risk bound: Since there exist worst-case instances where learning is hard, it is natural to expect that NN-LIFT has guarantees only when certain conditions are met. We assume that the input has a regular continuous probability density function (pdf); see (4.8) for the details. This is a reasonable assumption, since under Boolean inputs (a special case of discrete input), it reduces to learning parity with noise which is a hard problem [105]. We assume that the activating functions are *sufficiently non-linear*, since if they are linear, then the network can be collapsed into a single layer [33], which is non-identifiable. We precisely characterize how the estimation error depends on the non-linearity of the activating function through its third order derivative.

Another condition for providing the risk bound is non-redundancy of the neurons. If the neurons are redundant, it is an over-specified network. In the realizable setting, where the target function is generated by a neural network with the given number of neurons k, we require (tensorizations of)

the weights of first layer to be linearly independent. In the non-realizable setting, we require this to be satisfied by k vectors randomly drawn from the Fourier magnitude distribution (weighted by the norm of frequency vector) of the target function $f(x)$. More precisely, the random frequencies are drawn from probability distribution $\Lambda(\omega) := \|\omega\| \cdot |F(\omega)|/C_f$ where $F(\omega)$ is the Fourier transform of arbitrary function $f(x)$, and C_f is the normalization factor; see (4.23) and corresponding discussions for more details. This is a mild condition which holds when the distribution is continuous in some domain. Thus, our conditions for achieving bounded risk are mild and encompass a large class of target functions and input distributions.

Why tensors are required? We employ the cross-moment tensor which encodes the correlation between the third order score function and the output. We then decompose the moment tensor as a sum of rank-1 components to yield the weight vectors of the first layer. We require at least a third order tensor to learn the neural network weights for the following reasons: while a matrix decomposition is only identifiable up to orthogonal components, tensors can have identifiable non-orthogonal components. In general, it is not realistic to assume that the weight vectors are orthogonal, and hence, we require tensors to learn the weight vectors. Moreover, through tensors, we can learn overcomplete networks, where the number of hidden neurons can exceed the input/output dimensions. Note that matrix factorization methods are unable to learn overcomplete models, since the rank of the matrix cannot exceed its dimensions. Thus, it is critical to incorporate tensors for training neural networks. A recent set of papers have analyzed the tensor methods in detail, and established convergence and perturbation guarantees [20, 18, 22, 17, 41], despite non-convexity of the decomposition problem. Such strong theoretical guarantees are essential for deriving provable risk bounds for NN-LIFT.

Extensions: Our algorithm NN-LIFT can be extended to more layers, by recursively estimating the weights layer by layer. In principle, our analysis can be extended by controlling the perturbation introduced due to layer-by-layer estimation. Establishing precise guarantees is an exciting open problem.

In this work, we assume knowledge of the generative model for the input. As argued before, in many settings such as experimental design or polling, the design of the input pdf $p(x)$ is under

the control of the learner. Even if $p(x)$ is not known, a recent flurry of research activity has shown that a wide class of probabilistic models can be trained consistently using a suite of different efficient algorithms: convex relaxation methods [51], spectral and tensor methods [18], alternating minimization [4], and they require only polynomial sample and computational complexity, with respect to the input and hidden dimensions. These methods can learn a rich class of models which also includes latent or hidden variable models.

Another aspect not addressed in this work is the issue of regularization for our NN-LIFT algorithm. In this work, we assume that the number of neurons is chosen appropriately to balance bias and variance through cross validation. Designing implicit regularization methods such as *dropout* [90] or *early stopping* [128] for tensor factorization and analyzing them rigorously is another exciting open research problem.

4.2 Related works

We first review some works regarding the analysis of backpropagation, and then provide some theoretical results on training neural networks.

Analysis of backpropagation and loss surface of optimization: Baldi and Hornik [33] show that if the activations are linear, then backpropagation has a unique local optimum, and it corresponds to the principal components of the covariance matrix of the training examples. However, it is known that there exist networks with non-linear activations where backpropagation fails; for instance, Brady et al. [46] construct simple cases of linearly separable classes that backpropagation fails. Note that the simple perceptron algorithm will succeed here due to linear separability. Gori and Tesi [79] argue that such examples are artificial and that backpropagation succeeds in reaching the global optimum for linearly separable classes in practical settings. However, they show that under non-linear separability, backpropagation can get stuck in local optima. For a detailed survey, see [75].

Recently, Choromanska et al. [55] analyze the loss surface of a multi-layer ReLU network by relating it to a spin glass system. They make several assumptions such as variable independence for the input, equally likely paths from input to output, redundancy in network parameterization and uniform distribution for unique weights, which are far from realistic. Under these assumptions, the network reduces to a random *spin glass model*, where it is known that the lowest critical values of the random loss function form a layered structure, and the number of local minima outside that band diminishes exponentially with the network size [32]. However, this does not imply computational efficiency: there is no guarantee that we can find such a good local optimal point using computationally cheap algorithms, since there are still exponential number of such points.

Haeffele and Vidal [82] provide a general framework for characterizing when local optima become global in deep learning and other scenarios. The idea is that if the network is sufficiently overspecified (i.e., has enough hidden neurons) such that there exist local optima where some of the neurons have zero contribution, then such local optima are in fact, global. This provides a simple and a unified characterization of local optima which are global. However, in general, it is not clear how to design algorithms that can reach these efficient optimal points.

Previous theoretical works for training neural networks: Analysis of risk for neural networks is a classical problem. Approximation error of two layer neural network has been analyzed in a number of works [64, 93, 36]. Barron [36] provides a bound on the approximation error and combines it with the estimation error to obtain a risk bound, but for a computationally inefficient method. The sample complexity for neural networks have been extensively analyzed in [36, 38], assuming convergence to the globally optimal solution, which in general is intractable. See Anthony and Bartlett [25], Shalev-Shwartz and Ben-David [141] for an exposition of classical results on neural networks.

Andoni et al. [24] learn polynomial target functions using a two-layer neural network under Gaussian/uniform input distribution. They argue that the weights for the first layer can be selected randomly, and only the second layer weights, which are linear, need to be fitted optimally. However, in practice, Gaussian/uniform distributions are never encountered in classification problems. For general distributions, random weights in the first layer is not sufficient. Under our framework, we

impose only mild non-degeneracy conditions on the weights. Livni et al. [120] make the observation that networks with quadratic activation functions can be trained in a computationally efficient manner in an incremental manner. This is because with quadratic activations, greedily adding one neuron at a time can be solved efficiently through eigen decomposition. However, the standard sigmoidal networks require a large depth polynomial network, which is not practical. After we posted the initial version of this chapter, Zhang et al. [161] extended this framework to improper learning scenario, where the output predictor need not be a neural network. They show that if the ℓ_1 norm of the incoming weights in each layer is bounded, then learning is efficient. However, for the usual neural networks with sigmoidal activations, the ℓ_1 norm of the weights scales with dimension and in this case, the algorithm is no longer polynomial time. Arora et al. [31] provide bounds for leaning a class of deep representations. They use layer-wise learning where the neural network is learned layer-by-layer in an unsupervised manner. They assume sparse edges with random bounded weights, and 0/1 threshold functions in hidden nodes. The difference is here, we are considering the supervised setting where there is both input and output, and we allow for general sigmoidal functions at the hidden neurons.

Recently, after posting the initial version of this chapter, Hardt et al. [85] provided an analysis of stochastic gradient descent and its generalization error in convex and non-convex problems such as training neural networks. They show that the generalization error can be controlled under mild conditions. However, their work does not address about reaching a solution with small risk bound using SGD, and the SGD in general can get stuck in a spurious local optima. On the other hand, we show that in addition to having a small generalization error, our method yields a neural network with a small risk bound. Note that our method is moment-based estimation, and these methods come with stability bounds that guarantee good generalization error.

Closely related to this work, Sedghi and Anandkumar [140] consider learning neural networks with sparse connectivity. They employ the cross-moment between the (multi-class) label and (first order) score function of the input. They show that they can provably learn the weights of the first layer, as long as the weights are sparse enough, and there are enough number of input dimensions and output classes (at least linear up to log factor in the number of neurons in any layer). In this

chapter, we remove these restrictions and allow for the output to be just binary class (and indeed, our framework applies for multi-class setting as well, since the amount of information increases with more label classes from the algorithmic perspective), and for the number of neurons to exceed the input/output dimensions (overcomplete setting). Moreover, we extend beyond the realizable setting, and do not require the target functions to be generated from the class of neural networks under consideration.

4.3 Preliminaries and Problem Formulation

We first introduce some notations and then propose the problem formulation.

Let $[n] := \{1, 2, \ldots, n\}$, and $\|u\|$ denote the ℓ_2 or Euclidean norm of vector u, and $\langle u, v\rangle$ denote the inner product of vectors u and v. For matrix $C \in \mathbb{R}^{d\times k}$, the j-th column is referred by C_j or c_j, $j \in [k]$. Throughout this chapter, $\nabla_x^{(m)}$ denotes the m-th order derivative operator w.r.t. variable x. For matrices $A, B \in \mathbb{R}^{d\times k}$, the *Khatri-Rao product* $C := A \odot B \in \mathbb{R}^{d^2\times k}$ is defined such that $C(l + (i-1)d, j) = A(i, j) \cdot B(l, j)$, for $i, l \in [d], j \in [k]$.

Derivative: For function $g(x) : \mathbb{R}^d \to \mathbb{R}$ with vector input $x \in \mathbb{R}^d$, the m-th order derivative w.r.t. variable x is denoted by $\nabla_x^{(m)} g(x) \in \bigotimes^m \mathbb{R}^d$ (which is a m-th order tensor) such that

$$\left[\nabla_x^{(m)} g(x)\right]_{i_1,\ldots,i_m} := \frac{\partial g(x)}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_m}}, \quad i_1, \ldots, i_m \in [d]. \tag{4.1}$$

When it is clear from the context, we drop the subscript x and write the derivative as $\nabla^{(m)} g(x)$.

Fourier transform: For a function $f(x) : \mathbb{R}^d \to \mathbb{R}$, the multivariate Fourier transform $F(\omega) : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$F(\omega) := \int_{\mathbb{R}^d} f(x) e^{-j\langle \omega, x\rangle} dx, \tag{4.2}$$

where variable $\omega \in \mathbb{R}^d$ is called the frequency variable, and j denotes the imaginary unit. We also denote the Fourier pair $(f(x), F(\omega))$ as $f(x) \xleftrightarrow{\text{Fourier}} F(\omega)$.

Function notations: Throughout the chapter, we use the following convention to distinguish different types of functions. We use $f(x)$ (or y) to denote an arbitrary function and exploit $\tilde{f}(x)$ (or $\tilde{y}$) to denote the output of a realizable neural network. This helps us to differentiate between them. We also use notation $\widehat{f}(x)$ (or $\widehat{y}$) to denote the estimated (trained) neural networks using finite number of samples.

4.3.1 Problem formulation

We now introduce the problem of training a neural network in realizable and non-realizable settings, and elaborate on the notion of risk bound on how the trained neural network approximates an arbitrary function. It is known that continuous functions with compact domain can be arbitrarily well approximated by feedforward neural networks with one hidden layer and sigmoidal nonlinear functions [63, 92, 35].

The input (feature) is denoted by variable x, and output (label) is denoted by variable y. We assume the input and output are generated according to some joint density function $p(x, y)$ such that $(x_i, y_i) \overset{\text{i.i.d.}}{\sim} p(x, y)$, where (x_i, y_i) denotes the i-th sample. We assume knowledge of the input density $p(x)$, and demonstrate how it can be used to train a neural network to approximate the conditional density $p(y|x)$ in a computationally efficient manner. We discuss in Section 4.4.1 how the input density $p(x)$ can be estimated through numerous methods such as score matching or spectral methods. In settings such as experimental design, the input density $p(x)$ is known to the learner since she designs the density function, and our framework is directly applicable there. In addition, we do not need to know the normalization factor or the partition function of the input distribution $p(x)$, and the estimation up to normalization factor suffices.

Risk bound: In this chapter, we propose a new algorithm for training neural networks and provide risk bounds with respect to an arbitrary target function. Risk is the expected loss over the joint probability density function of input x and output y. Here, we consider the squared ℓ_2 loss and bound the *risk error*

$$\mathbb{E}_x[|f(x) - \widehat{f}(x)|^2], \tag{4.3}$$

where $f(x)$ is an arbitrary function which we want to approximate by $\widehat{f}(x)$ denoting the estimated (trained) neural network. This notion of risk is also called mean integrated squared error. The proposed risk error for a neural network consists of two parts: approximation error and estimation error. *Approximation error* is the error in fitting the target function $f(x)$ to a neural network with the given architecture $\tilde{f}(x)$, and *estimation error* is the error in training that network with finite number of samples denoted by $\widehat{f}(x)$. Thus, the risk error measures the ability of the trained neural network to generalize to new data generated by function $f(x)$. We now introduce the realizable and non-realizable settings, which elaborates more these sources of error.

4.3.1.1 Realizable setting

In the realizable setting, the output is generated by a neural network. We consider a neural network with one hidden layer of dimension k. Let the output $\tilde{y} \in \{0, 1\}$ be the binary label, and $x \in \mathbb{R}^d$ be the feature vector; see Section 4.7.2 for generalization to higher dimensional output (multi-label and multi-class), and also the continuous output case. We consider the label generating model

$$\tilde{f}(x) := \mathbb{E}[\tilde{y}|x] = \langle a_2, \sigma(A_1^\top x + b_1)\rangle + b_2, \tag{4.4}$$

where $\sigma(\cdot)$ is (linear/nonlinear) elementwise function. See Figure 4.1 for a schematic representation of label-function in (4.4) in the general case of vector output $\tilde{y}$.

In the realizable setting, the goal is to train the neural network in (4.4), i.e., to learn the weight matrices (vectors) $A_1 \in \mathbb{R}^{d\times k}$, $a_2 \in \mathbb{R}^k$ and bias vectors $b_1 \in \mathbb{R}^k$, $b_2 \in \mathbb{R}$. This only involves the estimation analysis where we have a label-function $\tilde{f}(x)$ specified in (4.4) with fixed unknown parameters A_1, b_1, a_2, b_2, and the goal is to learn these parameters and finally bound the overall function estimation error $\mathbb{E}_x[|\tilde{f}(x) - \widehat{f}(x)|^2]$, where $\widehat{f}(x)$ is the estimation of fixed neural network $\tilde{f}(x)$ given finite samples. For this task, we propose a computationally efficient algorithm which requires only polynomial number of samples for bounded estimation error. This is the first time that such a result has been established for any neural network.

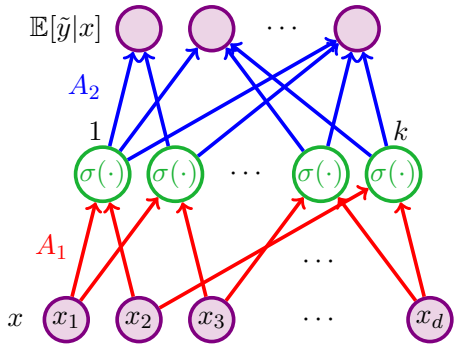


Figure 4.1: Graphical representation of a neural network, $\mathbb{E}[\tilde{y}|x] = A_2^\top \sigma(A_1^\top x + b_1) + b_2$. Note that this representation is for general vector output $\tilde{y}$ which can be also written as $\mathbb{E}[\tilde{y}|x] = \langle a_2, \sigma(A_1^\top x + b_1)\rangle + b_2$ in the case of scalar output $\tilde{y}$.

4.3.1.2 Non-realizable setting

In the non-realizable setting, the output is an arbitrary function $f(x)$ which is not necessarily a neural network. We want to approximate $f(x)$ by $\widehat{f}(x)$ denoting the estimated (trained) neural network. In this setting, the additional approximation analysis is also required. In this chapter, we combine the estimation result in realizable setting with the approximation bounds in Barron [35] leading to risk bounds with respect to the target function $f(x)$; see (4.3) for the definition of risk. The detailed results are provided in Section 4.6.

4.4 NN-LIFT Algorithm

In this section, we introduce our proposed method for learning neural networks using tensor, Fourier and regression techniques. Our method is shown in Algorithm 6 named NN-LIFT (Neural Network LearnIng using Feature Tensors). The algorithm has three main components. The first component involves estimating the weight matrix of the first layer denoted by $A_1 \in \mathbb{R}^{d\times k}$ by a tensor decomposition method. The second component involves estimating the bias vector of the first layer $b_1 \in \mathbb{R}^k$ by a Fourier method. We finally estimate the parameters of last layer $a_2 \in \mathbb{R}^k$ and $b_2 \in \mathbb{R}$ by linear regression.

Procedure 6 NN-LIFT (Neural Network LearnIng using Feature Tensors)

input Labeled samples $\{(x_i, y_i) : i \in [n]\}$, parameter $\tilde{\epsilon}_1$, parameter λ.
input Third order score function $\mathcal{P}_3(x)$ of the input x; see Equation (4.5) for the definition.
1: Compute $\widehat{T} := \frac{1}{n}\sum_{i\in[n]} y_i \cdot \mathcal{P}_3(x_i)$.
2: $\{(\hat{A}_1)_j\}_{j\in[k]}$ = tensor decomposition($\widehat{T}$); see Section 4.4.2 and Appendix D.1 for details.
3: $\widehat{b}_1$ = Fourier method($\{(x_i, y_i) : i \in [n]\}, \widehat{A}_1, \tilde{\epsilon}_1$); see Procedure 7.
4: $(\widehat{a}_2, \widehat{b}_2)$ = Ridge regression($\{(x_i, y_i) : i \in [n]\}, \widehat{A}_1, \widehat{b}_1, \lambda$); see Procedure 8.
5: **return** $\hat{A}_1, \widehat{a}_2, \widehat{b}_1, \widehat{b}_2$.

Note that most of the unknown parameters (compare the dimensions of matrix A_1, vectors a_2, b_1, and scalar b_2) are estimated in the first part, and thus, this is the main part of the algorithm. Given this fact, we also provide an alternative method for the estimation of other parameters of the model, given the estimate of A_1 from the tensor method. This is based on incrementally adding neurons, one by one, whose first layer weights are given by A_1 and the remaining parameters are updated using brute force search on a grid. Since each update involves just updating the corresponding bias term b_1, and its contribution to the final output, this is low dimensional, and can be done efficiently; details are in Section 4.7.3.

We now explain the steps of Algorithm 6 in more details.

4.4.1 Score function

The m-th order score function $\mathcal{P}_m(x) \in \bigotimes^m \mathbb{R}^d$ is defined as [102]

$$\mathcal{P}_m(x) := (-1)^m \frac{\nabla_x^{(m)} p(x)}{p(x)}, \tag{4.5}$$

where $p(x)$ is the probability density function of random vector $x \in \mathbb{R}^d$. In addition, $\nabla_x^{(m)}$ denotes the m-th order derivative operator; see (4.1) for the precise definition. The main property of score functions as yielding differential operators that enables us to estimate the weight matrix A_1 via tensor decomposition is discussed in the next subsection; see Equation (4.6).

In this work, we assume access to a sufficiently good approximation of the input pdf $p(x)$ and the corresponding score functions $\mathcal{S}_2(x)$, $\mathcal{S}_3(x)$. Indeed, estimating these quantities in general is a

hard problem, but there exist numerous instances where this becomes tractable. Examples include spectral methods for learning latent variable models such as Gaussian mixtures, topic or admixture models, independent component analysis (ICA) and so on [18]. Moreover, there have been recent advances in non-parametric score matching methods [150] for density estimation in infinite dimensional exponential families with guaranteed convergence rates. These methods can be used to estimate the input pdf in an unsupervised manner. Below, we discuss in detail about score function estimation methods. In this work, we focus on how we can use the input generative information to make training of neural networks tractable. For simplicity, in the subsequent analysis, we assume that these quantities are perfectly known; it is possible to extend the perturbation analysis to take into account the errors in estimating the input pdf; see Remark 14.

Estimation of score function: There are various efficient methods for estimating the score function. The framework of score matching is popular for parameter estimation in probabilistic models [99, 152], where the criterion is to fit parameters based on matching the data score function. Swersky et al. [152] analyze the score matching for latent energy-based models. In deep learning, the framework of auto-encoders attempts to find encoding and decoding functions which minimize the reconstruction error under added noise; the so-called Denoising Auto-Encoders (DAE). This is an unsupervised framework involving only unlabeled samples. Alain and Bengio [5] argue that the DAE approximately learns the first order score function of the input, as the noise variance goes to zero. Sriperumbudur et al. [150] propose non-parametric score matching methods for density estimation in infinite dimensional exponential families with guaranteed convergence rates. Therefore, we can use any of these methods for estimating $\mathcal{P}_1(x)$ and use the recursive form [102]

$$\mathcal{P}_m(x) = -\mathcal{P}_{m-1}(x) \otimes \nabla_x \log p(x) - \nabla_x \mathcal{P}_{m-1}(x)$$

to estimate higher order score functions.

4.4.2 Tensor decomposition

The score functions are new representations (extracted features) of input data x that can be used for training neural networks. We use score functions and labels of training data to form the empirical cross-moment $\widehat{T} = \frac{1}{n}\sum_{i\in[n]} y_i \cdot \mathcal{P}_3(x_i)$. We decompose tensor $\widehat{T}$ to estimate the columns of A_1. The following discussion reveals why tensor decomposition is relevant for this task.

The score functions have the property of yielding differential operators with respect to the input distribution. More precisely, for label-function $f(x) := \mathbb{E}[y|x]$, Janzamin et al. [102] show that

$$\mathbb{E}[y \cdot \mathcal{S}_3(x)] = \mathbb{E}[\nabla_x^{(3)} f(x)].$$

Now for the neural network output in (4.4), note that the function $\tilde{f}(x)$ is a non-linear function of both input x and weight matrix A_1. The expectation operator $\mathbb{E}[\cdot]$ averages out the dependency on x, and the derivative acts as a *linearization operator* as follows. In the neural network output (4.4), we observe that the columns of weight vector A_1 are the linear coefficients involved with input variable x. When taking the derivative of this function, by the chain rule, these linear coefficients shows up in the final form. In particular, we show in Lemma 4.1 (see Section 4.8) that for neural network in (4.4), we have

$$\mathbb{E}\left[\tilde{y} \cdot \mathcal{P}_3(x)\right] = \sum_{j\in[k]} \lambda_j \cdot (A_1)_j \otimes (A_1)_j \otimes (A_1)_j \in \mathbb{R}^{d\times d\times d}, \tag{4.6}$$

where $(A_1)_j \in \mathbb{R}^d$ denotes the j-th column of A_1, and $\lambda_j \in \mathbb{R}$ denotes the coefficient; refer to Equation (1.5) for the notion of tensor rank and its rank-1 components. This clarifies how the score function acts as a linearization operator while the final output is nonlinear in terms of A_1. The above form also clarifies the reason behind using tensor decomposition in the learning framework.

Tensor decomposition algorithm: The goal of tensor decomposition algorithm is to recover the rank-1 components of tensor. For this step, we use the tensor decomposition algorithm proposed in Section 2.3; see Appendix D.1 for more details. Here, we first orthogonalize the tensor via whitening

procedure and then apply the tensor power iteration. Thus, the original tensor decomposition need not to be orthogonal.

Computational Complexity: It is popular to perform the tensor decomposition in a stochastic manner which reduces the computational complexity. This is done by splitting the data into mini-batches. Starting with the first mini-batch, we perform a small number of tensor power iterations, and then use the result as initialization for the next mini-batch, and so on. As mentioned earlier, we assume that a sufficiently good approximation of score function tensor is given to us. For specific cases where we have this tensor in factor form, we can reduce the computational complexity of NN-LIFT by not computing the whole tensor explicitly. By having factor form, we mean when we can write the score function tensor in terms of summation of rank-1 components which could be the summation over samples, or from other existing structures in the model. We now state a few examples when we have the factor form, and provide the computational complexity. For example, if input follows Gaussian distribution, the score function has a simple polynomial form, and the computational complexity of tensor decomposition is $O(nkdR)$, where n is the number of samples and R is the number of initializations for the tensor decomposition. Similar argument follows when the input distribution is mixture of Gaussian distributions.

We can also analyze complexity for more complex inputs. If we fit the input data into a Restricted Boltzmann Machines (RBM) model, the computational complexity of our method is $O(nkdd_hR)$. Here d_h is the number of neurons of the first layer of the RBM used for approximating the input distribution. Tensor methods are also embarrassingly parallelizable. When performed in parallel, the computational complexity would be $O(\log(\min\{d, d_h\}))$ with $O(nkdd_hR/\log(\min(d, d_h)))$ processors. Alternatively, we can also exploit recent tensor sketching approaches [158] for computing tensor decompositions efficiently. Wang et al. [158] build on the idea of count sketches and show that the running time is linear in the input dimension and the number of samples, and is independent in the order of the tensor. Thus, tensor decompositions can be computed efficiently.

Procedure 7 Fourier method for estimating b_1

input Labeled samples $\{(x_i, y_i) : i \in [n]\}$, estimate $\widehat{A}_1$, parameter $\tilde{\epsilon}_1$.
input Probability density function $p(x)$ of the input x.

1: **for** $l = 1$ to k **do**
2: Let $\Omega_l := \left\{\omega \in \mathbb{R}^d : \|\omega\| = \frac{1}{2}, \left|\langle \omega, (\widehat{A}_1)_l \rangle\right| \geq \frac{1-\tilde{\epsilon}_1^2/2}{2}\right\}$, and $|\Omega_l|$ denotes the surface area of $d-1$ dimensional manifold Ω_l.
3: Draw n i.i.d. random frequencies $\omega_i, i \in [n]$, uniformly from set Ω_l.
4: Let $v := \frac{1}{n}\sum_{i\in[n]} \frac{y_i}{p(x_i)} e^{-j\langle \omega_i, x_i \rangle}$ which is a complex number as $v = |v|e^{j\angle v}$. The operators $|\cdot|$ and $\angle\cdot$ respectively denote the magnitude and phase operators.
5: Let $\widehat{b}_1(l) := \frac{1}{\pi}(\angle v - \angle\Sigma(1/2))$, where $\sigma(x) \xleftrightarrow{\text{Fourier}} \Sigma(\omega)$.
6: **return** $\widehat{b}_1$.

4.4.3 Fourier method

The second part of the algorithm estimates the first layer bias vector $b_1 \in \mathbb{R}^k$. This step is very different from the previous step for estimating A_1 which was based on tensor decomposition methods. This is a Fourier-based method where complex variables are formed using labeled data and random frequencies in the Fourier domain; see Procedure 7. We prove in Lemma 4.6 that the entries of b_1 can be estimated from the phase of these complex numbers. We also observe in Lemma 4.6 that the magnitude of these complex numbers can be used to estimate a_2; this is discussed in Appendix D.2.2.

Polynomial-time random draw from set Ω_l: Note that the random frequencies are drawn from a $d-1$ dimensional manifold denoted by Ω_l which is the intersection of sphere $\|\omega\| = \frac{1}{2}$ and cone $\left|\langle \omega, (\widehat{A}_1)_l \rangle\right| \geq \frac{1-\tilde{\epsilon}_1^2/2}{2}$ in $\mathbb{R}^d$. This manifold is actually the surface of a spherical cap. In order to draw these frequencies in polynomial time, we consider the d-dimensional spherical coordinate system such that one of the angles is defined based on the cone axis. We can then directly impose the cone constraint by limiting the corresponding angle in the random draw. In addition, Kothari and Meka [109] propose a method for generating pseudo-random variables from the spherical cap in logarithmic time.

Remark 11 (Knowledge of input distribution only up to normalization factor). The computation of score function and the Fourier method both involve knowledge about input pdf $p(x)$. However, we do not need to know the normalization factor, also known as partition function, of the input pdf. For the score function, it is immediately seen from the definition in (4.5) since the normalization

Procedure 8 Ridge regression method for estimating a_2 and b_2

input Labeled samples $\{(x_i, y_i) : i \in [n]\}$, estimates $\widehat{A}_1$ and $\widehat{b}_1$, regularization parameter λ.

1: Let $\widehat{h}_i := \sigma(\widehat{A}_1^\top x_i + \widehat{b}_1)$, $i \in [n]$, denote the estimation of the neurons.

2: Append each neuron $\widehat{h}_i$ by the dummy variable 1 to represent the bias, and thus, $\widehat{h}_i \in \mathbb{R}^{k+1}$.

3: Let $\hat{\Sigma}_{\widehat{h}} := \frac{1}{n}\sum_{i\in[n]} \widehat{h}_i \widehat{h}_i^\top \in \mathbb{R}^{(k+1)\times(k+1)}$ denote the empirical covariance of $\widehat{h}$.

4: Let $\hat{\beta}_\lambda \in \mathbb{R}^{k+1}$ denote the estimated parameters by λ-regularized ridge regression as

$$\hat{\beta}_\lambda = \left(\hat{\Sigma}_{\widehat{h}} + \lambda I_{k+1}\right)^{-1} \cdot \frac{1}{n}\left(\sum_{i\in[n]} y_i \widehat{h}_i\right), \tag{4.7}$$

where I_{k+1} denotes the $(k+1)$-dimensional identity matrix.

5: **return** $\widehat{a}_2 := \hat{\beta}_\lambda(1:k)$, $\widehat{b}_2 := \hat{\beta}_\lambda(k+1)$.

factor is canceled out by the division by $p(x)$, and thus, the estimation of score function is at most as hard as estimation of input pdf up to normalization factor. In the Fourier method, we can use the non-normalized estimation of input pdf which leads to a normalization mismatch in the estimation of corresponding complex number. This is not a problem since we only use the phase information of these complex numbers.

4.4.4 Ridge regression method

For the neural network model in (4.4), given a good estimation of neurons, we can estimate the parameters of last layer by linear regression. We provide Procedure 8 in which we use ridge regression algorithm to estimate the parameters of last layer a_2 and b_2. See Appendix D.2.3 for the details of ridge regression and the corresponding analysis and guarantees.

4.5 Risk Bound in the Realizable Setting

In this section, we provide guarantees in the realizable setting, where the function $\tilde{f}(x) := \mathbb{E}[\tilde{y}|x]$ is generated by a neural network as in (4.4). We provide the estimation error bound on the overall function recovery $\mathbb{E}_x[|\tilde{f}(x) - \widehat{f}(x)|^2]$ when the estimation is done by Algorithm 6.

We provide guarantees in the following settings. 1) In the basic case, we consider the *undercomplete* regime $k \leq d$, and provide the results assuming A_1 is full column rank. 2) In the second case, we form higher order cross-moments and tensorize it into a lower order tensor. This enables us to learn the network in the overcomplete regime $k > d$, when the Khatri-Rao product $A_1 \odot A_1 \in \mathbb{R}^{d^2 \times k}$ is full column rank. We call this the *overcomplete* setting and this can handle up to $k = O(d^2)$. Similarly, we can extend to larger k by tensorizing higher order moments in the expense of additional computational complexity.

We define the following quantity for label function $\tilde{f}(\cdot)$ as

$$\tilde{\zeta}_{\tilde{f}} := \int_{\mathbb{R}^d} \tilde{f}(x) dx.$$

Note that in the binary classification setting ($\tilde{y} \in \{0, 1\}$), we have $\mathbb{E}[\tilde{y}|x] := \tilde{f}(x) \in [0, 1]$ which is always positive, and there is no square of $\tilde{f}(x)$ considered in the above quantity.

Let η denote the noise in the neural network model in (4.4) such that the output is

$$\tilde{y} = \tilde{f}(x) + \eta.$$

Note that the noise η is not necessarily independent of x; for instance, in the classification setting or binary output $\tilde{y} \in \{0, 1\}$, the noise in dependent on x.

Conditions for Theorem 4.3:

- **Non-degeneracy of weight vectors**: In the undercomplete setting ($k \leq d$), the weight matrix $A_1 \in \mathbb{R}^{d \times k}$ is full column rank and $s_{\min}(A_1) > \epsilon$, where $s_{\min}(\cdot)$ denotes the minimum singular value, and $\epsilon > 0$ is related to the target error in recovering the columns of A_1. In the overcomplete setting ($k \leq d^2$), the Khatri-Rao product $A_1 \odot A_1 \in \mathbb{R}^{d^2 \times k}$ is full column rank[3], and $s_{\min}(A_1 \odot A_1) > \epsilon$; see Remark 13 for generalization.

[3]It is shown in Bhaskara et al. [41] that this condition is satisfied under smoothed analysis.

- **Conditions on nonlinear activating function** $\sigma(\cdot)$: the coefficients

$$\lambda_j := \mathbb{E}\left[\sigma'''(z_j)\right] \cdot a_2(j), \quad \tilde{\lambda}_j := \mathbb{E}\left[\sigma''(z_j)\right] \cdot a_2(j), \quad j \in [k],$$

in (4.17) and (D.11) are nonzero. Here, $z := A_1^\top x + b_1$ is the input to the nonlinear operator $\sigma(\cdot)$. In addition, $\sigma(\cdot)$ satisfies the *Lipschitz* property[4] with constant L such that $|\sigma(u) - \sigma(u')| \leq L \cdot |u - u'|$, for $u, u' \in \mathbb{R}$. Suppose that the nonlinear activating function $\sigma(z)$ satisfies the property such that $\sigma(z) = 1 - \sigma(-z)$. Many popular activating functions such as step function, sigmoid function and tanh function satisfy this last property.

- **Subgaussian noise**: There exists a finite $\sigma_{\text{noise}} \geq 0$ such that, almost surely,

$$\mathbb{E}_\eta[\exp(\alpha\eta)|x] \leq \exp(\alpha^2\sigma_{\text{noise}}^2/2), \quad \forall \alpha \in \mathbb{R},$$

where η denotes the noise in the output $\tilde{y}$.

- **Bounded statistical leverage**: There exists a finite $\rho_\lambda \geq 1$ such that, almost surely,

$$\frac{\sqrt{k}}{\sqrt{(\inf\{\lambda_j\} + \lambda)k_{1,\lambda}}} \leq \rho_\lambda,$$

where $k_{1,\lambda}$ denotes the effective dimensions of the hidden layer $h := \sigma(A_1^\top x + b_1)$ as $k_{1,\lambda} := \sum_{j\in[k]} \frac{\lambda_j}{\lambda_j + \lambda}$. Here, λ_j's denote the (positive) eigenvalues of the hidden layer covariance matrix Σ_h, and λ is the regularization parameter of ridge regression.

We now elaborate on these conditions. The *non-degeneracy of weight vectors* are required for the tensor decomposition analysis in the estimation of A_1. In the undercomplete setting, the algorithm first orthogonalizes (through whitening procedure) the tensor given in (4.6), and then decomposes it through tensor power iteration. Note that the convergence of power iteration for orthogonal tensor decomposition is guaranteed [160, 18]. For the orthogonalization procedure to work, we need the tensor components (the columns of matrix A_1) to be linearly independent. In

[4] If the step function $\sigma(u) = 1_{\{u>0\}}(u)$ is used as the activating function, the Lipschitz property does not hold because of the non-continuity at $u = 0$. But, we can assume the Lipschitz property holds in the linear continuous part, i.e., when $u, u' > 0$ or $u, u' < 0$. We then argue that the input to the step function $1_{\{u>0\}}(u)$ is w.h.p. in the linear interval (where the Lipschitz property holds).

the overcomplete setting, the algorithm performs the same steps with the additional tensorizing procedure; see Appendix D.1 for details. In this case, a higher order tensor is given to the algorithm and it is first tensorized before performing the same steps as in the undercomplete setting. Thus, the same conditions are now imposed on $A_1 \odot A_1$.

In addition to the non-degeneracy condition on weight matrix A_1, the *coefficients condition* on λ_j's is also required to ensure the corresponding rank-1 components in (4.6) do not vanish, and thus, the tensor decomposition algorithm recovers them. Similarly, the coefficients $\tilde{\lambda}_j$ should be also nonzero to enable us using the second order moment $\tilde{M}_2$ in (D.10) in the whitening step of tensor decomposition algorithm. If one of the coefficients vanishes, we use the other option to perform the whitening; see Remark 15 and Procedure 10 for details. Note that the amount of non-linearity of $\sigma(\cdot)$ affects the magnitude of the coefficients. It is also worth mentioning that although we use the third derivative notation $\sigma'''(\cdot)$ in characterizing the coefficients λ_j (and similarly $\sigma''(\cdot)$ in $\tilde{\lambda}_j$), we do not need the differentiability of non-linear function $\sigma(\cdot)$ in all points. In particular, when input x is a continuous variable, we use Dirac delta function $\delta(\cdot)$ as the derivative in non-continuous points; for instance, for the derivative of step function $1_{\{x>0\}}(x)$, we have $\frac{d}{dx}1_{\{x>0\}}(x) = \delta(x)$. Thus, in general, we only need the expectations $\mathbb{E}\left[\sigma'''(z_j)\right]$ and $\mathbb{E}\left[\sigma''(z_j)\right]$ to exist for these type of functions and the corresponding higher order derivatives.

We impose the *Lipschitz* condition on the non-linear activating function to limit the error propagated in the hidden layer, when the first layer parameters are estimated by the neural network and Fourier methods. The condition $\sigma(z) = 1 - \sigma(-z)$ is also assumed to tackle the sign issue in recovering the columns of A_1; see Remark 29 for the details. The *subgaussian noise* and the *bounded statistical leverage* conditions are standard conditions, required for ridge regression, which is used for estimating the parameters of the second layer of the neural network. Both parameters σ_{noise}, and ρ_λ affect the sample complexity in the final guarantees.

Imposing additional bounds on the parameters of the neural network are useful in learning these parameters with computationally efficient algorithms since it limits the searching space for training these parameters. In particular, for the Fourier analysis, we assume the following conditions. Suppose the columns of weight matrix A_1 are normalized, i.e., $\|(A_1)_j\| = 1, j \in [k]$, and the entries

of first layer bias vector b_1 are bounded as $|b_1(l)| \leq 1, l \in [k]$. Note that the normalization condition on the columns of A_1 is also needed for identifiability of the parameters. For instance, if the non-linear operator is the step function $\sigma(z) = 1_{\{z>0\}}(z)$, then matrix A_1 is only identifiable up to its norm, and thus, such normalization condition is required for identifiability. The estimation of entries of the bias vector b_1 is obtained from the phase of a complex number through Fourier analysis; see Procedure 7 for details. Since there is ambiguity in the phase of a complex number[5], we impose the bounded assumption on the entries of b_1 to avoid this ambiguity.

Let $p(x)$ satisfy some mild regularity conditions on the boundaries of the support of $p(x)$. In particular, all the entries of (matrix-output) functions

$$\tilde{f}(x) \cdot \nabla^{(2)} p(x), \quad \nabla \tilde{f}(x) \cdot \nabla p(x)^\top, \quad \nabla^{(2)} \tilde{f}(x) \cdot p(x) \tag{4.8}$$

should go to zero on the boundaries of support of $p(x)$. These regularity conditions are required for the properties of the score function to hold; see Janzamin et al. [102] for more details.

In addition to the above main conditions, we also need some mild conditions which are not crucial for the recovery guarantees and are mostly assumed to simplify the presentation of the main results. These conditions can be relaxed more. Suppose the input x is bounded, i.e., $x \in B_r$, where $B_r := \{x : \|x\| \leq r\}$. Assume the input probability density function $p(x) \geq \psi$ for some $\psi > 0$, and for any $x \in B_r$. The regularity conditions in (4.8) might seem contradictory with the lower bound condition $p(x) \geq \psi$, but there is an easy fix for that. The lower bound on $p(x)$ is required for the analysis of the Fourier part of the algorithm. We can have a continuous $p(x)$, while in the Fourier part, we only use x's such that $p(x) \geq \psi$, and ignore the rest. This only introduces a probability factor $\Pr[x : p(x) \geq \psi]$ in the analysis.

Settings of algorithm in Theorem 4.3:

- No. of iterations in Algorithm 12: $N = \Theta\left(\log \frac{1}{\epsilon}\right)$.
- No. of initializations in Procedure 2: $R \geq \text{poly}(k)$.
- Parameter $\tilde{\epsilon}_1 = \tilde{O}\left(\frac{1}{\sqrt{n}}\right)$ in Procedure 7, where n is the number of samples.

[5] A complex number does not change if an integer multiple of 2π is added to its phase.

- We exploit the empirical second order moment $\widehat{M}_2 := \frac{1}{n}\sum_{i\in[n]} y_i \cdot \mathcal{P}_2(x_i)$, in the whitening Procedure 10, which is the first option stated in the procedure. See Remark 15 for further discussion about the other option.

Theorem 4.3 (NN-LIFT guarantees: estimation bound in the realizable setting)**.** *Assume the above settings and conditions hold. For $\epsilon > 0$, suppose the number of samples n satisfies (up to* log *factors)*

$$n \geq \tilde{O}\left(\frac{k}{\epsilon^2}\cdot \mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right] \cdot \text{poly}\left(\tilde{y}_{\max}, \frac{\mathbb{E}\left[\left\|\mathcal{S}_2(x)\mathcal{S}_2^\top(x)\right\|\right]}{\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]}, \frac{\tilde{\zeta}_{\tilde{f}}}{\psi}, \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}, \frac{1}{\lambda_{\min}}, \frac{s_{\max}(A_1)}{s_{\min}(A_1)}, |\Omega_l|, L, \frac{\|a_2\|}{(a_2)_{\min}}, |b_2|, \sigma_{noise}, \rho_\lambda\right)\right). \tag{4.9}$$

See (D.3), (D.26), (D.28) *and* (D.30) *for the complete form of sample complexity. Here, $M_3(x) \in \mathbb{R}^{d\times d^2}$ denotes the matricization of score function tensor $\mathcal{S}_3(x) \in \mathbb{R}^{d\times d\times d}$; see* (1.1) *for the definition of matricization. Furthermore, $\lambda_{\min} := \min_{j\in[k]}|\lambda_j|$, $\tilde{\lambda}_{\min} := \min_{j\in[k]}|\tilde{\lambda}_j|$, $\tilde{\lambda}_{\max} := \max_{j\in[k]}|\tilde{\lambda}_j|$, $(a_2)_{\min} := \min_{j\in[k]}|a_2(j)|$, and $\tilde{y}_{\max}$ is such that $|\tilde{f}(x)| \leq \tilde{y}_{\max}$, for $x \in B_r$. Then the function estimate $\widehat{f}(x) := \langle \widehat{a}_2, \sigma(\widehat{A}_1^\top x + \widehat{b}_1)\rangle + \widehat{b}_2$ using the estimated parameters $\widehat{A}_1, \widehat{b}_1, \widehat{a}_2, \widehat{b}_2$ (output of NN-LIFT Algorithm 6) satisfies the estimation error*

$$\mathbb{E}_x[|\widehat{f}(x) - \tilde{f}(x)|^2] \leq \tilde{O}(\epsilon^2).$$

See Section 4.8 and Appendix D.2 for the proof of theorem. Thus, we estimate the neural network in polynomial time and sample complexity. This is one of the first results to provide a guaranteed method for training neural networks with efficient computational and statistical complexity. Note that although the sample complexity in [35] is smaller as $n \geq \tilde{O}\left(\frac{kd}{\epsilon^2}\right)$, the proposed algorithm in [35] is not computationally efficient.

Remark 12 (Sample complexity for Gaussian input). If the input x is Gaussian as $x \sim \mathcal{N}(0, I_d)$, then we know that $\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right] = \tilde{O}\left(d^3\right)$ and $\mathbb{E}\left[\left\|\mathcal{S}_2(x)\mathcal{S}_2^\top(x)\right\|\right] = \tilde{O}\left(d^2\right)$, and the above sample complexity is simplified.

Remark 13 (Higher order tensorization). We stated that by tensorizing higher order tensors to lower order ones, we can estimate overcomplete models where the hidden layer dimension k is larger than

the input dimension d. We can generalize this idea to higher order tensorizing such that m modes of the higher order tensor are tensorized into a single mode in the resulting lower order tensor. This enables us to estimate the models up to $k = O(d^m)$ assuming the matrix $A_1 \odot \cdots \odot A_1$ (m Khatri-Rao products) is full column rank. This is possible with the higher computational complexity.

Remark 14 (Effect of erroneous estimation of $p(x)$). The input probability density function $p(x)$ is directly used in the Fourier part of the algorithm, and also indirectly used in the tensor decomposition part to compute the score function $\mathcal{S}_3(x)$; see (4.5). In the above analysis, to simplify the presentation, we assume we exactly know these functions, and thus, there is no additional error introduced by estimating them. It is straightforward to incorporate the corresponding errors in estimating input density into the final bound.

Remark 15 (Alternative whitening prodecure). In whitening Procedure 10, two options are provided for constructing the second order moment M_2. In the above analysis, we used the first option which exploits the second order score function. If any coefficient $\tilde{\lambda}_j, j \in [k]$, in (D.10) vanishes, we cannot use the second order score function in the whitening procedure, and we use the other option for whitening; see Procedure 10 for the details.

4.6 Risk Bound in the Non-realizable Setting

In this section, we provide the risk bound for training the neural network with respect to an arbitrary target function; see Section 4.3.1 for the definition of the risk.

In order to provide the risk bound with respect to an arbitrary target function, we also need to argue the approximation error in addition to the estimation error. For an arbitrary function $f(x)$, we need to find a neural network whose error in approximating the function can be bounded. We then combine it with the estimation error in training that neural network. This yields the final risk bound.

The approximation problem is about finding a neural network that approximates an arbitrary function $f(x)$ with bounded error. Thus, this is different from the realizable setting where there is a fixed neural network and we only analyze its estimation. Barron [35] provides an approximation

bound for the two-layer neural network and we exploit that here. His result is based on the Fourier properties of function $f(x)$. Recall from (4.2) the definition of Fourier transform of $f(x)$, denoted by $F(\omega)$, where ω is called the frequency variable. Define the first absolute moment of the Fourier magnitude distribution as

$$C_f := \int_{\mathbb{R}^d} \|\omega\|_2 \cdot |F(\omega)| d\omega. \tag{4.10}$$

Barron [35] analyzes the approximation properties of

$$\tilde{f}(x) = \sum_{j\in[k]} a_2(j)\sigma\big(\langle (A_1)_j, x\rangle + b_1(j)\big), \quad \|(A_1)_j\| = 1, |b_1(j)| \le 1, |a_2(j)| \le 2C_f, j \in [k], \tag{4.11}$$

where the columns of weight matrix A_1 are the normalized version of random frequencies drawn from the Fourier magnitude distribution $|F(\omega)|$ weighted by the norm of the frequency vector. More precisely,

$$\omega_j \overset{\text{i.i.d.}}{\sim} \frac{\|\omega\|}{C_f}|F(\omega)|, \quad (A_1)_j = \frac{\omega_j}{\|\omega_j\|}, \quad j \in [k]. \tag{4.12}$$

See Section 4.8.2.1 for a detailed discussion on this connection between the columns of weight matrix A_1 and the random frequency draws from the Fourier magnitude distribution, and see how this is argued in the proof of the approximation bound. The other parameters a_2, b_1 need to be also found. He then shows the following approximation bound for (4.11).

Theorem 4.4 (Approximation bound, Theorem 3 of Barron [35])**.** *For a function $f(x)$ with bounded C_f, there exists an approximation $\tilde{f}(x)$ in the form of* (4.11) *that satisfies the approximation bound*

$$\mathbb{E}_x[|\overline{f}(x) - \tilde{f}(x)|^2] \le O(r^2 C_f^2) \cdot \left(\frac{1}{\sqrt{k}} + \delta_1\right)^2,$$

where $\overline{f}(x) = f(x) - f(0)$. Here, for $\tau > 0$,

$$\delta_\tau := \inf_{0<\xi\le 1/2} \left\{ 2\xi + \sup_{|z|\ge\xi} \big|\sigma(\tau z) - 1_{\{z>0\}}(z)\big| \right\} \tag{4.13}$$

is a distance between the unit step function $1_{\{z>0\}}(z)$ and the scaled sigmoidal function $\sigma(\tau z)$.

See Barron [35] for the complete proof of the above theorem. For completeness, we have also reviewed the main ideas of this proof in Section 4.8.2. We now provide the formal statement of our risk bound.

Conditions for Theorem 4.5:

- The nonlinear activating function $\sigma(\cdot)$ is an arbitrary sigmoidal function satisfying the aforementioned Lipschitz condition. Note that a sigmoidal function is a bounded measurable function on the real line for which $\sigma(z) \to 1$ as $z \to \infty$ and $\sigma(z) \to 0$ as $z \to -\infty$.
- For $\epsilon > 0$, suppose the number of samples n satisfies (up to log factors)

$$n \geq \tilde{O}\left(\frac{k}{\epsilon^2} \cdot \mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]\right. \tag{4.14}$$
$$\left. \cdot \operatorname{poly}\left(y_{\max}, \frac{\mathbb{E}\left[\left\|\mathcal{S}_2(x)\mathcal{S}_2^\top(x)\right\|\right]}{\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]}, \frac{\zeta_f}{\psi}, \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}, \frac{1}{\lambda_{\min}}, \frac{s_{\max}(A_1)}{s_{\min}(A_1)}, \right.\right.$$
$$\left.\left. |\Omega_l|, L, \frac{\|a_2\|}{(a_2)_{\min}}, |b_2|, \sigma_{\text{noise}}, \rho_\lambda\right)\right),$$

  where $\zeta_f := \int_{\mathbb{R}^d} f(x)^2 dx$; notice the difference with $\tilde{\zeta}_{\tilde{f}}$. Note that this is the same sample complexity as in Theorem 4.3 with $\tilde{y}_{\max}$ substituted with $y_{\max}$ and $\tilde{\zeta}_{\tilde{f}}$ substituted with ζ_f.

- The target function $f(x)$ is bounded, and for $\epsilon > 0$, it has bounded C_f as

$$C_f \leq \tilde{O}\left(\left(\frac{1}{\sqrt{k}} + \delta_1\right)^{-1} \cdot \left(\frac{1}{\sqrt{k}} + \epsilon\right) \cdot \frac{1}{\sqrt{\mathbb{E}\left[\|\mathcal{S}_3(x)\|^2\right]}}\right. \tag{4.15}$$
$$\left. \cdot \operatorname{poly}\left(\frac{1}{r}, \frac{\mathbb{E}\left[\|\mathcal{S}_3(x)\|^2\right]}{\mathbb{E}\left[\|\mathcal{S}_2(x)\|^2\right]}, \psi, \frac{\tilde{\lambda}_{\min}}{\tilde{\lambda}_{\max}}, \lambda_{\min}, \frac{s_{\min}(A_1)}{s_{\max}(A_1)}, \right.\right.$$
$$\left.\left. \frac{1}{|\Omega_l|}, \frac{1}{L}, \frac{(a_2)_{\min}}{\|a_2\|}, \frac{1}{|b_2|}, \frac{1}{\sigma_{\text{noise}}}, \frac{1}{\rho_\lambda}\right)\right).$$

  See (D.31) and (D.33) for the complete form of bound on C_f. For Gaussian input $x \sim \mathcal{N}(0, I_d)$, we have $\sqrt{\mathbb{E}\left[\|\mathcal{S}_3(x)\|^2\right]} = \tilde{O}\left(d^{1.5}\right)$, and $r = \tilde{O}(\sqrt{d})$.

  See Corollary 4.1 for examples of functions that satisfy this bound, and thus, we can learn them by the proposed method.

- The coefficients $\lambda_j := \mathbb{E}\left[\sigma'''(z_j)\right] \cdot a_2(j)$, and $\tilde{\lambda}_j := \mathbb{E}\left[\sigma''(z_j)\right] \cdot a_2(j)$, $j \in [k]$, in (4.17) and (D.11) are non-zero.
- k random i.i.d. draws of frequencies in Equation (4.12) are linearly independent. Note that the draws are from Fourier magnitude distribution[6] $\|\omega\| \cdot |F(\omega)|$. For more discussions on this condition, see Section 4.8.2.1 and earlier explanations in this section. In the overcomplete regime, $(k > d)$, the linear independence property needs to hold for appropriate tensorizations of the frequency draws.

The above requirements on the number of samples n and parameter C_f depend on the parameters of the neural network A_1, a_2, b_1 and b_2. Note that there is also a dependence on these parameters through coefficients λ_j and $\tilde{\lambda}_j$. Since this is the non-realizable setting, these neural network parameters correspond to the neural networks that satisfy the approximation bound proposed in Theorem 4.4 and are generated via random draws from the frequency spectrum of the function $f(x)$.

The proposed bound on C_f in (4.15) is stricter when the number of hidden units k increases. This might seem counter-intuitive, since the approximation result in Theorem 4.4 suggests that increasing k leads to smaller approximation error. But, note that the approximation result in Theorem 4.4 does not consider efficient training of the neural network. The result in Theorem 4.5 also deals with the efficient estimation of the neural network. This imposes additional constraint on the parameter C_f such that when the number of neurons increases, the problem of learning the network weights is more challenging for the tensor method to resolve.

Theorem 4.5 (NN-LIFT guarantees: risk bound)**.** *Suppose the above conditions hold. Then the target function f is approximated by the neural network $\widehat{f}$ which is learnt using NN-LIFT in Algorithm 6 satisfying w.h.p.*

$$\mathbb{E}_x[|f(x) - \widehat{f}(x)|^2] \leq O(r^2 C_f^2) \cdot \left(\frac{1}{\sqrt{k}} + \delta_1\right)^2 + O(\epsilon^2),$$

where δ_τ is defined in (4.13)*. Recall $x \in B_r$, where $B_r := \{x : \|x\| \leq r\}$.*

[6]Note that it should be normalized to be a probability distribution as in (4.12).

The theorem is mainly proved by combining the estimation bound guarantees in Theorem 4.3, and the approximation bound results for neural networks provided in Theorem 4.4. But note that the approximation bound provided in Theorem 4.4 holds for a specific class of neural networks which are not generally recovered by the NN-LIFT algorithm. In addition, the estimation guarantees in Theorem 4.3 is for the realizable setting where the observations are the outputs of a fixed neural network, while in Theorem 4.5 we observe samples of arbitrary function $f(x)$. Thus, the approximation analysis in Theorem 4.4 can not be directly applied to Theorem 4.3. For this, we need additional assumptions to ensure the NN-LIFT algorithm recovers a neural network which is close to one of the neural networks that satisfy the approximation bound in Theorem 4.4. Therefore, we impose the bound on quantity C_f, and the full column rank assumption proposed in Theorem 4.4. See Appendix D.3 for the complete proof of Theorem 4.5.

The above risk bound includes two terms. The first term $O(r^2 C_f^2) \cdot \left(\frac{1}{\sqrt{k}} + \delta_1\right)^2$ represents the approximation error on how the arbitrary function $f(x)$ with quantity C_f can be approximated by the neural network, whose weights are drawn from the Fourier magnitude distribution; see Theorem 4.4 for the formal statement. From the definition of C_f in (4.10), this bound is weaker when the Fourier spectrum of target $f(x)$ has more energy in higher frequencies. This makes intuitive sense since it should be easier to approximate a function which is more smooth and has less fluctuations. The second term $O(\epsilon^2)$ is from estimation error for NN-LIFT algorithm, which is analyzed in Theorem 4.3. The polynomial factors for sample complexity in our estimation error are slightly worse than the bound provided in Barron [36], but note that we provide an estimation method which is both computationally and statistically efficient, while the method in Barron [36] is not computationally efficient. Thus, for the first time, we have a computationally efficient method with guaranteed risk bounds for training neural networks.

Discussion on δ_τ in the approximation bound: The approximation bound involves a term δ_τ which is a constant and does not shrink with increasing the neuron size k. Recall that δ_τ measures the distance between the unit step function $1_{\{z>0\}}(z)$ and the scaled sigmoidal function $\sigma(\tau z)$ (which is used in the neural network specified in (4.4)). We now provide the following two observations

The above risk bound is only provided for the case $\tau = 1$. We can generalize this result by imposing different constraint on the norm of columns of A_1 in (4.11). In general, if we impose $\|(A_1)_j\| = \tau, j \in [k]$, for some $\tau > 0$, then we have the approximation bound[7] $O(r^2 C_f^2) \cdot \left(\frac{1}{\sqrt{k}} + \delta_\tau\right)^2$. Note that $\delta_\tau \to 0$ when $\tau \to \infty$ (the scaled sigmoidal function $\sigma(\tau z)$ converges to the unit step function), and thus, this constant approximation error vanishes.

If the sigmoidal function is the unit step function as $\sigma(z) = 1_{\{z>0\}}(z)$, then $\delta_\tau = 0$ for all $\tau > 0$, and hence, there is no such constant approximation error.

4.7 Discussions and Extensions

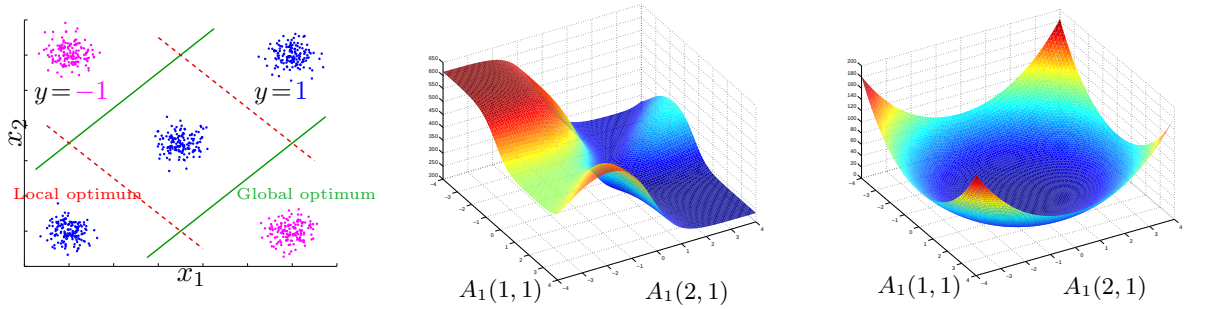
In this section, we provide additional discussions. We first propose a toy example contrasting the hardness of optimization problems backpropagation and tensor decomposition. We then discuss the generalization of learning guarantees to higher dimensional output, and also the continuous output case. We then discuss an alternative approach for estimating the low-dimensional parameters of the model.

4.7.1 Contrasting the loss surface of backpropagation with tensor decomposition

We discussed that the computational hardness of training a neural network is due to the non-convexity of the loss function, and thus, popular local search methods such as backpropagation can get stuck in spurious local optima. We now provide a toy example highlighting this issue, and contrast it with the tensor decomposition approach.

We consider a simple binary classification task shown in Figure 4.2.a, where blue and magneta data points correspond to two different classes. It is clear that these two classes can be classified by a mixture of two linear classifiers which are drawn as green solid lines in the figure. For this task, we consider a two-layer neural network with two hidden neurons. The loss surfaces for backpropagation

[7] Note that this change also needs some straightforward appropriate modifications in the algorithm.



(a) Classification setup

(b) Loss surface for backprop.

(c) Loss surface for tensor method

Figure 4.2: (a) Classification task: two colors correspond to binary labels. A two-layer neural network with two hidden neurons is used. Loss surface in terms of the first layer weights of one neuron (i.e., weights connecting the inputs to the neuron) is plotted while other parameters are fixed. (b) Loss surface for usual square loss objective has spurious local optima. In part (a), one of the spurious local optima is drawn as red dashed lines and the global optimum is drawn as green solid lines. (c) Loss surface for tensor factorization objective is free of spurious local optima.

and tensor decomposition are shown in Figures 4.2.b and 4.2.c, respectively. They are shown in terms of the weight parameters of inputs to the first neuron, i.e., the first column of matrix A_1, while the weight parameters to the second neuron are randomly drawn, and then fixed.

The stark contrast between the optimization landscape of tensor objective function, and the usual square loss objective used for backpropagation are observed, where even for a very simple classification task, backpropagation suffers from spurious local optima (one set of them is drawn as red dashed lines), which is not the case with tensor methods that is at least locally convex. This comparison highlights the algorithmic advantage of tensor decomposition compared to backpropagation in terms of the optimization they are performing.

4.7.2 Extensions to cases beyond binary classification

We earlier limited ourselves to the case where the output of neural network $\tilde{y} \in \{0, 1\}$ is binary. These results can be easily extended to more complicated cases such as higher dimensional output (multi-label and multi-class), and also the continuous outputs (i.e., regression setting). In the rest of this section, we discuss about the necessary changes in the algorithm to adapt it for these cases.

In the multi-dimensional case, the output label $\tilde{y}$ is a vector generated as

$$\mathbb{E}[\tilde{y}|x] = A_2^\top \sigma(A_1^\top x + b_1) + b_2,$$

where the output is either discrete (multi-label and multi-class) or continuous. Recall that the algorithm includes three main parts: tensor decomposition, Fourier and ridge regression components.

Tensor decomposition: For the tensor decomposition part, we first form the empirical version of $\tilde{T} = \mathbb{E}\left[\tilde{y} \otimes \mathcal{S}_3(x)\right]$; note that $\otimes$ is used here (instead of scalar product used earlier) since $\tilde{y}$ is not a scalar anymore. By the properties of score function, this tensor has decomposition form

$$\tilde{T} = \mathbb{E}\left[\tilde{y} \otimes \mathcal{P}_3(x)\right] = \sum_{j\in[k]} \mathbb{E}\left[\sigma'''(z_j)\right] \cdot (A_2)^j \otimes (A_1)_j \otimes (A_1)_j \otimes (A_1)_j,$$

where $(A_2)^j$ denotes the j^{th} row of matrix A_2. This is proved similar to Lemma 4.1. The tensor $\tilde{T}$ is a fourth order tensor, and we contract the first mode by multiplying it with a random vector θ as $\tilde{T}(\theta, I, I, I)$ leading to the same form in (4.16) as

$$\tilde{T}(\theta, I, I, I) = \sum_{j\in[k]} \lambda_j \cdot (A_1)_j \otimes (A_1)_j \otimes (A_1)_j,$$

with λ_j changed to $\lambda_j = \mathbb{E}\left[\sigma'''(z_j)\right] \cdot \langle (A_2)^j, \theta\rangle$. Therefore, the same tensor decomposition guarantees in the binary case also hold here when the empirical version of $\tilde{T}(\theta, I, I, I)$ is the input to the algorithm.

Fourier method: Similar to the scalar case, we can use one of the entries of output to estimate the entries of b_1. There is an additional difference in the continuous case. Suppose that the output is generated as $\tilde{y} = \tilde{f}(x) + \eta$ where η is noise vector which is independent of input x. In this case, the parameter $\tilde{\zeta}_{\tilde{f}}$ corresponding to l^{th} entry of output $\tilde{y}_l$ is changed to $\tilde{\zeta}_{\tilde{f}} := \int_{\mathbb{R}^d} \tilde{f}(x)_l^2 dx + \int_{\mathbb{R}} \eta_l^2 dt$.

Ridge regression: The ridge regression method and analysis can be immediately generalized to non-scalar output by applying the method independently to different entries of output vector to recover different columns of matrix A_2 and different entries of vector b_2.

4.7.3 An alternative for estimating low-dimensional parameters

Once we have an estimate of the first layer weights A_1, we can greedily (i.e., incrementally) add neurons with the weight vectors $(A_1)_j$ for $j \in [k]$, and choose the bias $b_1(j)$ through grid search, and learn its contribution $a_2(j)$ for its final output. This is on the lines of the method proposed in Barron [35], with one crucial difference that in our case, the first layer weights A_1 are already estimated by the tensor method. Barron [35] proposes optimizing for each weight vector $(A_1)_j$ in d-dimensional space, whose computational complexity can scale exponentially in d in the worst case. But, in our setup here, since we have already estimated the high-dimensional parameters (i.e., the columns of A_1), we only need to estimate a few low dimensional parameters. For the new hidden unit indexed by j, these parameters include the bias from input layer to the neuron (i.e., $b_1(j)$), and the weight from the neuron to the output (i.e., $a_2(j)$). This makes the approach computationally tractable, and we can even use brute-force or exhaustive search to find the best parameters on a finite set and get guarantees akin to [35].

4.8 Proof Sketch

In this section, we provide key ideas for proving the main results in Theorems 4.3 and 4.4.

4.8.1 Estimation bound

The estimation bound is proposed in Theorem 4.3, and the complete proof is provided in Appendix D.2. Recall that NN-LIFT algorithm includes a tensor decomposition part for estimating A_1, a Fourier technique for estimating b_1, and a linear regression for estimating a_2, b_2. The application of linear regression in the last layer is immediately clear. In this section, we propose two main lemmas which clarify why the other methods are useful for estimating the unknown parameters A_1, b_1 in the realizable setting, where the label $\tilde{y}$ is generated by the neural network with the given architecture.

In the following lemma, we show how the cross-moment between label and score function as $\mathbb{E}[\tilde{y} \cdot \mathcal{S}_3(x)]$ leads to a tensor decomposition form for estimating weight matrix A_1.

Lemma 4.1. *For the two-layer neural network specified in* (4.4)*, we have*

$$\mathbb{E}\left[\tilde{y} \cdot \mathcal{P}_3(x)\right] = \sum_{j \in [k]} \lambda_j \cdot (A_1)_j \otimes (A_1)_j \otimes (A_1)_j, \tag{4.16}$$

where $(A_1)_j \in \mathbb{R}^d$ *denotes the* j*-th column of* A_1*, and*

$$\lambda_j = \mathbb{E}\left[\sigma'''(z_j)\right] \cdot a_2(j), \tag{4.17}$$

for vector $z := A_1^\top x + b_1$ *as the input to the nonlinear operator* $\sigma(\cdot)$*.*

This is proved by the main property of score functions as yielding differential operators, where for label-function $f(x) := \mathbb{E}[y|x]$, we have $\mathbb{E}[y \cdot \mathcal{S}_3(x)] = \mathbb{E}[\nabla_x^{(3)} f(x)]$ [102]; see Section D.2.1 for a complete proof of the lemma. This lemma shows that by decomposing the cross-moment tensor $\mathbb{E}[\tilde{y} \cdot \mathcal{S}_3(x)]$, we can recover the columns of A_1.

We also exploit the phase of complex number v to estimate the bias vector b_1; see Procedure 7. The following lemma clarifies this. The perturbation analysis is provided in the appendix.

Lemma 4.6. *Let*

$$\tilde{v} := \frac{1}{n} \sum_{i \in [n]} \frac{\tilde{y}_i}{p(x_i)} e^{-j\langle \omega_i, x_i \rangle}. \tag{4.18}$$

Notice this is a realizable of v *in Procedure 7 where the output corresponds to a neural network* $\tilde{y}$*. If* ω_i*'s are uniformly i.i.d. drawn from set* Ω_l*, then* $\tilde{v}$ *has mean (which is computed over* x*,* $\tilde{y}$ *and* ω*)*

$$\mathbb{E}[\tilde{v}] = \frac{1}{|\Omega_l|} \Sigma\left(\frac{1}{2}\right) a_2(l) e^{j\pi b_1(l)}, \tag{4.19}$$

where $|\Omega_l|$ *denotes the surface area of* $d-1$ *dimensional manifold* Ω_l*, and* $\Sigma(\cdot)$ *denotes the Fourier transform of* $\sigma(\cdot)$*.*

This lemma is proved in Appendix D.2.2.

4.8.2 Approximation bound

We exploit the approximation bound argued in Barron [35] provided in Theorem 4.4. We first discuss his main result arguing an approximation bound $O(r^2C_f^2/k)$ for a function $f(x)$ with bounded parameter C_f; see (4.10) for the definition of C_f. Note that this corresponds to the first term in the approximation error proposed in Theorem 4.4. For this result, Barron [35] does not consider any bound on the parameters of first layer A_1 and b_1. He then provides a refinement of this result where he also bounds the parameters of neural network as we also do in (4.11). This leads to the additional term involving δ_τ in the approximation error as seen in Theorem 4.4. Note that bounding the parameters of neural network is also useful in learning these parameters with computationally efficient algorithms since it limits the searching space for training these parameters. We now provide the main ideas of proving these bounds as follows.

4.8.2.1 No bounds on the parameters of the neural network

We first provide the proof outline when there is no additional constraints on the parameters of neural network; see set G defined in (4.21), and compare it with the form we use in (4.11) where there are additional bounds. In this case, Barron [35] argues approximation bound $O(r^2C_f^2/k)$ which is proved based on two main results. The first result says that if a function f is in the closure of the convex hull of a set G in a Hilbert space, then for every $k \geq 1$, there is an f_k as the convex combination of k points in G such that

$$\mathbb{E}[|f - f_k|^2] \leq \frac{c'}{k}, \tag{4.20}$$

for any constant c' satisfying some lower bound related to the properties of set G and function f; see Lemma 1 in Barron [35] for the precise statement and the proof of this result.

The second part of the proof is to argue that arbitrary function $f \in \Gamma$ (where Γ denotes the set of functions with bounded C_f) is in the closure of the convex hull of sigmoidal functions

$$G := \big\{\gamma\sigma\big(\langle\alpha, x\rangle + \beta\big) : \alpha \in \mathbb{R}^d, \beta \in \mathbb{R}, |\gamma| \leq 2C\big\}. \tag{4.21}$$

Barron [35] proves this result by arguing the following chain of inclusions as

$$\Gamma \subset \mathrm{cl}\, G_{\cos} \subset \mathrm{cl}\, G_{\text{step}} \subset \mathrm{cl}\, G,$$

where $\mathrm{cl}\, G$ denotes the closure of set G, and sets $G_{\cos}$ and G_{step} respectively denote set of some sinusoidal and step functions. See Theorem 2 in Barron [35] for the precise statement and the proof of this result.

Random frequency draws from Fourier magnitude distribution: Recall from Section 4.6 that the columns of weight matrix A_1 are the normalized version of random frequencies drawn from Fourier magnitude distribution $\|\omega\| \cdot |F(\omega)|$; see Equation (4.12). This connection is along the proof of relation $\Gamma \subset \mathrm{cl}\, G_{\cos}$ that we recap here; see proof of Lemma 2 in Barron [35] for more details. By expanding the Fourier transform as magnitude and phase parts $F(\omega) = e^{j\theta(\omega)}|F(\omega)|$, we have

$$\overline{f}(x) := f(x) - f(0) = \int g(x, \omega)\Lambda(d\omega), \tag{4.22}$$

where

$$\Lambda(\omega) := \|\omega\| \cdot |F(\omega)|/C_f \tag{4.23}$$

is the normalized Fourier magnitude distribution (as a probability distribution) weighted by the norm of frequency vector, and

$$g(x, \omega) := \frac{C_f}{\|\omega\|}\left(\cos(\langle\omega, x\rangle + \theta(\omega)) - \cos(\theta(\omega))\right).$$

The integral in (4.22) is an infinite convex combination of functions in the class

$$G_{\cos} := \left\{ \frac{\gamma}{\|\omega\|} \left(\cos(\langle \omega, x\rangle + \beta) - \cos(\beta)\right) : \omega \neq 0, |\gamma| \leq C, \beta \in \mathbb{R} \right\}.$$

Now if $\omega_1, \omega_2, \ldots, \omega_k$ is a random sample of k points independently drawn from Fourier magnitude distribution Λ, then by Fubini's Theorem, we have

$$\mathbb{E} \int_{B_r} \left(f(x) - \frac{1}{k} \sum_{j \in [k]} g(x, \omega_j) \right)^2 \mu(dx) \leq \frac{C^2}{k},$$

where $\mu(\cdot)$ is the probability measure for x. This shows function $\overline{f}$ is in the convex hull of $G_{\cos}$. Note that the bound $\frac{C^2}{k}$ complies the bound in (4.20).

4.8.2.2 Bounding the parameters of the neural network

Barron [35] then imposes additional bounds on the weights of first layer, considering the following class of sigmoidal functions as

$$G_\tau := \left\{ \gamma \sigma\big(\tau(\langle \alpha, x\rangle + \beta)\big) : \|\alpha\| \leq 1, |\beta| \leq 1, |\gamma| \leq 2C \right\}. \tag{4.24}$$

Note that the approximation proposed in (4.11) is a convex combination of k points in (4.24) with $\tau = 1$. Barron [35] concludes Theorem 4.4 by the following lemma.

Lemma 4.2 (Lemma 5 in Barron [35])**.** *If g is a function on $[-1, 1]$ with derivative bounded*[8] *by a constant C, then for every $\tau > 0$, we have*

$$\inf_{g_\tau \in \operatorname{cl} G_\tau} \sup_{|z| \leq \tau} |g(z) - g_\tau(z)| \leq 2C\delta_\tau.$$

Finally Theorem 4.4 is proved by applying triangle inequality to bounds argued in the above two cases.

[8]Note that the condition on having bounded derivative does not rule out cases such as step function as the sigmoidal function. This is because similar to the analysis for the main case (no bounds on the weights), we first argue that function f is in the closure of functions in $G_{\cos}$ which are univariate functions with bounded derivative.

Chapter 5

Identifiability of Overcomplete Topic Models: Uniqueness of Tensor Tucker Decomposition

Overcomplete latent representations have been very popular for unsupervised feature learning in recent years. In this chapter, we specify which overcomplete models can be identified given observable moments of a certain order. We consider probabilistic admixture or topic models in the overcomplete regime, where the number of latent topics can greatly exceed the size of the observed word vocabulary. While general overcomplete topic models are not identifiable, we establish *generic* identifiability under a constraint, referred to as *topic persistence*. Our sufficient conditions for identifiability involve a novel set of "higher order" expansion conditions on the *topic-word matrix* or the *population structure* of the model. This set of higher-order expansion conditions allows for overcomplete models, and require the existence of a perfect matching from latent topics to higher order observed words. We establish that random structured topic models are identifiable w.h.p. in the overcomplete regime. Our identifiability results allows for general (non-degenerate) distributions for modeling the topic proportions, and thus, we can handle arbitrarily correlated topics in our framework. Our identifiability results imply uniqueness of a class of tensor decompositions with

structured sparsity which is contained in the class of *Tucker* decompositions, but is more general than the *Candecomp/Parafac* (CP) decomposition.

The performance of many machine learning methods is hugely dependent on the choice of data representations or features. Overcomplete representations, where the number of features can be greater than the dimensionality of the input data, have been extensively employed, and are arguably critical in a number of applications such as speech and computer vision [40]. Overcomplete representations are known to be more robust to noise, and can provide greater flexibility in modeling [119]. Unsupervised estimation of overcomplete representations has been hugely popular due to the availability of large-scale unlabeled samples in many applications.

A probabilistic framework for incorporating features posits latent or hidden variables that can provide a good explanation to the observed data. Overcomplete probabilistic models can incorporate a much larger number of latent variables compared to the observed dimensionality. In this chapter, we characterize the conditions under which overcomplete latent variable models can be identified from their observed moments.

For any parametric statistical model, identifiability is a fundamental question of whether the model parameters can be uniquely recovered given the observed statistics. Identifiability is crucial in a number of applications where the latent variables are the quantities of interest, e.g. inferring diseases (latent variables) through symptoms (observations), inferring communities (latent variables) via the interactions among the actors in a social network (observations), and so on. Moreover, identifiability can be relevant even in predictive settings, where feature learning is employed for some higher level task such as classification. For instance, non-identifiability can lead to the presence of non-isolated local optima for optimization-based learning methods, and this can affect their convergence properties, e.g., see Uschmajew [154].

In this chapter, we characterize identifiability for a popular class of latent variable models, known as the *admixture* or *topic* models [43, 134]. These are hierarchical mixture models, which incorporate the presence of multiple latent states (i.e. topics) in each document consisting of a tuple of observed variables (i.e. words). Previous works have established that the model parameters can be estimated

efficiently using low order observed moments (second and third order) under some non-degeneracy assumptions, e.g. Anandkumar et al. [15], Anandkumar et al. [9], Arora et al. [30]. However, these non-degeneracy conditions imply that the model is undercomplete, i.e., the latent dimensionality (number of topics) cannot exceed the observed dimensionality (word vocabulary size). In this work, we remove this restriction and consider overcomplete topic models, where the number of topics can far exceed the word vocabulary size.

It is perhaps not surprising that general topic models are not identifiable in the overcomplete regime. To this end, we introduce an additional constraint on the model, referred to as *topic persistence*, which roughly means that topics (i.e. latent states) persist locally in a sequence of observed words (but not necessarily globally). This "locality" effect among the observed words is not present in the usual "bag-of-words" or *exchangeable* topic model. Such local dependencies among observations abound in applications such as text, images and speech, and can lead to a more faithful representation. In addition, we establish that the presence of topic persistence is central towards obtaining model identifiability in the overcomplete regime, and we provide an in-depth analysis of this phenomenon in this work.

5.1 Summary of Results

In this work, we provide conditions for *generic*[1] model identifiability of overcomplete topic models given observable moments of a certain order (i.e., having a certain number of words in each document). We introduce the notion of *topic persistence*, and analyze its effect on identifiability. We establish identifiability in the presence of a novel combinatorial object, referred to as *perfect n-gram matching*, in the bipartite graph from topics to words. Finally, we prove that random structured topic models satisfy these criteria, and are thus identifiable in the overcomplete regime.

[1] A model is generically identifiable, if all the parameters in the parameter space are identifiable, almost surely. Refer to Definition 5.1 for more discussion.

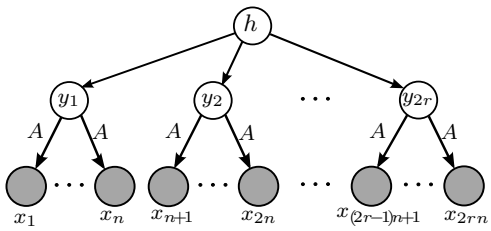


Figure 5.1: Hierarchical structure of the n-persistent topic model is illustrated for $2rn$ number of words (views) where $r \geq 1$ is an integer. A single topic $y_j, j \in [2r]$, is chosen for each sequence of n views $\{x_{(j-1)n+1}, \ldots, x_{(j-1)n+n}\}$. Matrix A is the population structure or topic-word matrix.

5.1.1 Persistent Topic Model

We first introduce the n-persistent topic model, where the parameter n determines the persistence level of a common topic in a sequence of n successive words. For instance, in Figure 5.1, the sequence of successive words $x_1, \ldots, x_n$ share a common topic y_1, and similarly, the words $x_{n+1}, \ldots, x_{2n}$ share topic y_2, and so on. The n-persistent model reduces to the popular "bag-of-words" model, when $n = 1$, and to the single topic model (i.e. only one topic in each document) when $n \to \infty$. Intuitively, topic persistence aids identifiability since we have multiple *views* of the common hidden topic generating a sequence of successive words. We establish that the bag-of-words model (with $n = 1$) is too non-informative about the topics in the overcomplete regime, and is therefore, not identifiable. On the other hand, n-persistent overcomplete topic models with $n \geq 2$ can become identifiable, and we establish a set of transparent conditions for identifiability.

5.1.2 Deterministic Conditions for Identifiability

Our sufficient conditions for identifiability are in the form of expansion conditions from the latent topic space to the observed word space. In the overcomplete regime, there are more topics than words in the vocabulary, and thus it is impossible to have expansion on the bipartite graph from topics to words, i.e., the graph encoding the sparsity pattern of the topic-word matrix. Instead, we impose an expansion constraint from topics to "higher order" words, which allows us to incorporate overcomplete models. We establish that this condition translates to the presence of a novel combinatorial object, referred to as the *perfect n-gram matching*, on the topic-word bipartite graph. Intuitively, the perfect n-gram matching condition implies "diversity" among the higher-order word

supports for different topics which leads to identifiability. In addition, we present trade-offs among the following quantities: number of topics, size of the word vocabulary, the topic persistence level, the order of the observed moments at hand, the minimum and maximum degrees of any topic in the topic-word bipartite graph, and the *Kruskal rank* [111] of the topic-word matrix, under which identifiability holds. To the best of our knowledge, this is the first work to provide conditions for characterizing identifiability of overcomplete topic models with structured sparsity.

As a corollary of our result, we also show that the expansion condition can be removed if the topic-word matrix is full column rank (and therefore undercomplete) and the model is persistent with persistence level at least two.

5.1.3 Identifiability of Random Structured Topic Models

We explicitly characterize the regime of identifiability for the random setting, where each topic i is supported on a random set of d_i words. Therefore, the bipartite graph from topics to words is a random graph with prescribed degrees for topics. For this random model with q topics, p-dimensional word vocabulary, and topic persistence level n, when $q = O(p^n)$ and $\Theta(\log p) \leq d_i \leq \Theta(p^{1/n})$, for all topics i, the topic-word matrix is identifiable from $2n^{\text{th}}$ order observed moments with high probability. Intuitively, the upper bound on the degrees d_i is needed to limit the overlap of word supports among different topics in the overcomplete regime: as the number of topics q increases (i.e., n increases in the above degree bound), the degree needs to be correspondingly smaller to ensure identifiability, and we make this dependence explicit. Intuitively, as the extent of overcompleteness increases, we need sparser connections from topics to words to ensure sufficient diversity in the word supports among different topics. The lower bound on the degrees is required so that there are enough edges in the topic-word bipartite graph so that various topics can be distinguished from one another. Furthermore, we establish that the size condition $q = O(p^n)$ for identifiability is tight.

As in the deterministic case, we also argue the result in the undercomplete setting and show that if $q \leq O(p)$ and $d_i \geq \Omega(\log p)$, then the topic-word matrix is identifiable from $2n^{\text{th}}$ order observed

moment with high probability under the persistent model with persistence level n at least equal to two. Here, the upper bound on the degree is relaxed and hence there is no sparsity constraints on the topic-word matrix.

5.1.4 Implications on Uniqueness of Overcomplete Tucker and CP Decompositions

We establish that identifiability of an overcomplete topic model is equivalent to uniqueness of decomposition of the observed moment tensor (of a certain order). Our identifiability results for persistent topic models imply uniqueness of a structured class of tensor decompositions, which is contained in the class of *Tucker* decompositions, but is more general than the *candecomp/parafac* (CP) decomposition [108]. This sub-class of Tucker decompositions involves structured sparsity and symmetry constraints on the *core tensor*, and sparsity constraints on the *inverse factors* of the Tucker decomposition. The structural constraints on the Tucker tensor decomposition are related to the topic model as follows: the sparsity and symmetry constraints on the core tensor are related to the persistence property of the topic model, and the sparsity constraints on the inverse factors are equivalent to the sparsity constraints on the topic-word matrix. For n-persistent topic model with $n = 1$ (bag-of-words model), the tensor decomposition is a general Tucker decomposition, where the core tensor is fully dense, while for $n \to \infty$ (single-topic model), the tensor decomposition reduces to a CP decomposition, i.e. the core tensor is a *diagonal tensor*. For a finite persistence level n, in between these two extremes, the core tensor satisfies certain sparsity and symmetry constraints, which becomes crucial towards establishing identifiability in the overcomplete regime.

5.2 Overview of Techniques

We now provide a short overview of the techniques employed in this work.

Recap of Identifiability Conditions in Under-complete Setting (Expansion Conditions on Topic-Word Matrix): Our approach is based on the recent results of Anandkumar et al. [9], where condi-

tions for identifiability of topic models are derived, given pairwise observed moments (specifically, co-occurrence of word-pairs in documents). Consider a topic model with q topics and observed word vocabulary of size p. Let $A \in \mathbb{R}^{p \times q}$ denote the topic-word matrix. Expansion conditions are imposed in Anandkumar et al. [9] on the topic-word bipartite graph which imply that (generically) the sparsest vectors in the column span of A, denoted by $\mathrm{Col}(A)$, are the columns of A themselves. Thus the topic-word matrix A is identifiable from pairwise moments under expansion constraints. However, these expansion conditions constrain the model to be under-complete, i.e., the number of topics $q \leq p$, the size of the word vocabulary. Therefore, the techniques derived in Anandkumar et al. [9] are not directly applicable here since we consider overcomplete models.

Identifiability in Overcomplete Setting and Why Topic-Persistence Helps: Pairwise moments are thus not sufficient for identifiability of overcomplete models, and the question is whether higher order moments can yield identifiability. We can view the higher order moments as pairwise moments of another equivalent topic model, which enables us to apply the techniques of Anandkumar et al. [9]. The key question is whether we have expansion in the equivalent topic model, which implies identifiability. For a general topic model (without any topic persistence constraints), it can be shown that for identifiability, we require expansion of the n^{th}-order *Kronecker product* of the original topic-word matrix A, denoted by $A^{\otimes n} \in \mathbb{R}^{p^n \times q^n}$, when given access to $(2n)^{\text{th}}$-order moments, for any integer $n \geq 1$. In the overcomplete regime where $q > p$, $A^{\otimes n}$ cannot expand, and therefore, overcomplete models are not identifiable in general. On the other hand, we show that imposing the constraint of topic persistence can lead to identifiability. For a n-persistent topic model, given $(2n)^{\text{th}}$-order moments, we establish that identifiability occurs when the n^{th}-order *Khatri-Rao product* of A, denoted by $A^{\odot n} \in \mathbb{R}^{p^n \times q}$, expands. Note that the Khatri-Rao product $A^{\odot n}$ is a sub-matrix of the Kronecker product $A^{\otimes n}$, and the Khatri-Rao product $A^{\odot n}$ can expand as long as $q \leq p^n$. Thus, the property of topic persistence is central towards achieving identifiability in the overcomplete regime.

First-Order Approach for Identifiability of Overcomplete Models (Expansion of n-gram Topic-Word Matrix): We refer to $A^{\odot n} \in \mathbb{R}^{p^n \times q}$ as the n-gram topic-word matrix, and intuitively, it relates topics to n-tuple words. Imposing the expansion conditions derived in Anandkumar et al. [9] on

$A^{\odot n}$ implies that (generically) the sparsest vectors in $\mathrm{Col}(A^{\odot n})$, are the columns of $A^{\odot n}$ themselves. Thus, the topic-word matrix A is identifiable from $(2n)^{\text{th}}$-order moments for a n-persistent topic model. We refer to this as the "first-order" approach since we directly impose the expansion conditions of Anandkumar et al. [9] on $A^{\odot n}$, without exploiting the additional structure present in $A^{\odot n}$.

Why the First-Order Approach is not Enough: Note that $A^{\odot n} \in \mathbb{R}^{p^n \times q}$ matrix relates topics to n-tuples of words. Thus, the entries of $A^{\odot n}$ are highly correlated, even if the original topic-word matrix A is assumed to be randomly generated. It is non-trivial to derive conditions on A, so that $A^{\odot n}$ expands. Moreover, we establish that $A^{\odot n}$ fails to expand on "small" sets, as required in [9], when the degrees are sufficiently different [2]. Thus, the first-order approach is highly restrictive in the overcomplete setting.

Incorporating Rank Criterion: Note that $A^{\odot n}$ is highly structured: the columns of $A^{\odot n}$ matrix possess a tensor [3] rank of 1, when $n > 1$. This can be incorporated in our identifiability criteria as follows: we provide conditions under which the sparsest vectors in $\mathrm{Col}(A^{\odot n})$, which also possess a tensor rank of 1, are the columns of $A^{\odot n}$ themselves. This implies identifiability of a n-persistent topic model, when given access to $(2n)^{\text{th}}$-order moments. Note that when a small number of columns of $A^{\odot n}$ are combined, the resulting vector cannot possess a tensor rank of 1, and thus, we can rule out that such sparse combinations of columns using the rank criterion. The maximum such number is at least the *Kruskal rank* [4] of A. Thus, sparse combinations of columns of A (up to the Kruskal rank) can be ruled out using the rank criterion, and we require expansion on $A^{\odot n}$ only on large sets of topics (of size larger than the Kruskal rank). This agrees with the intuition that when the topic-word matrix A has a larger Kruskal rank, it should be easier to identify A, since the Kruskal rank is related to the *mutual incoherence* [5] among the columns of A, see [76].

[2] For $A^{\odot n}$ to expand on a set of size $s \geq 2$, it is necessary that $s \cdot \binom{d_{\min}+n-1}{n} \geq s + \binom{d_{\max}+n-1}{n}$, where $d_{\min}$ and $d_{\max}$ are the minimum and maximum degrees, and n is the extent of overcompleteness: $q = \Theta(p^n)$. When the model is highly overcomplete (large n) and we require small set expansion (small s), the degrees need to be nearly the same. Thus, it is desirable to impose expansion only on large sets, since it allows for more degree diversity.

[3] When any column of $A^{\odot n} \in \mathbb{R}^{p^n \times q}$ (of length p^n) is reshaped as a n^{th}-order tensor $T \in \mathbb{R}^{p \times p \times \cdots \times p}$, the tensor T is rank 1.

[4] The Kruskal rank is the maximum number k such that every k-subset of columns of A are linearly independent. Note that the Kruskal rank is equal to the rank of A, when A has full column rank. But this cannot happen in the overcomplete setting.

[5] It is easy to show that $\mathrm{krank}(A) \geq (\max_{i \neq j} |a_i^\top a_j|)^{-1}$, where a_i, a_j are any pair of columns of A. Thus, higher incoherence leads to a larger kruskal rank.

Notion of Perfect n-gram Matching and Final Identifiability Conditions: Thus, we establish identifiability of overcomplete topic models subject to expansion conditions $A^{\odot n}$ on sets of size larger than the Kruskal rank of the topic-word matrix A. However, it is desirable to impose transparent and interpretable conditions directly on A for identifiability. We introduce the notion of *perfect n-gram matching* on the topic-word bipartite graph, which ensures that each topic can be uniquely matched to a n-tuple word. This combined with a lower bound on the Kruskal rank provides the final set of deterministic conditions for identifiability of the overcomplete topic model. Intuitively, we require that the columns of A be sparse, while still maintaining a large enough Kruskal rank; in other words, the topics have to be sparse and have sufficiently diverse word supports. Thus, we establish identifiability under a set of transparent conditions on the topic-word matrix A, consisting of perfect n-gram matching condition and a lower bound on the Kruskal rank of A.

Analysis under Random-Structured Topic-Word Matrices: Finally, we establish that the derived deterministic conditions are satisfied when the topic-word bipartite graph is randomly generated, as long as the degrees satisfy certain lower and upper bounds. Intuitively, a lower bound on the degrees of the topics is required to have degree concentration on various subsets so that expansion can occur, while the upper bound is required so that the Kruskal rank of the topic-word matrix is large enough compared to the sparsity level. Here, the main technical result is establishing the presence of a perfect n-gram matching in a random bipartite graph with a wide range of degrees. We present a greedy and a recursive mechanism for constructing such a n-gram matching for overcomplete models, which can be relevant even in other settings. For instance, our results imply the presence of a perfect matching when the edges of a bipartite graph are correlated in a structured manner, as given by the Khatri-Rao product.

5.3 Related Works

We now summarize some recent related works in the area of identifiability and learning of latent variable models.

5.3.0.1 Identifiability, Learning and Applications of Overcomplete Latent Representations

Many recent works employ unsupervised estimation of overcomplete features for higher level tasks such classification, e.g. [57, 118, 71, 40], and record huge gains over other approaches in a number of applications such as speech recognition and computer vision. However, theoretical understanding regarding learnability or identifiability of overcomplete representations is far more limited.

Overcomplete latent representations have been analyzed in the context of the independent components analysis (ICA), where the sources are assumed to be independent, and the mixing matrix is unknown. In the overcomplete or under-determined regime of the ICA, there are more sources than sensors. Identifiability and learning of the overcomplete ICA reduces to the problem of finding an overcomplete candecomp/parafac (CP) tensor decomposition. The classical result by Kruskal provides conditions for uniqueness of a CP decomposition [111, 112], with recent extensions to the notion of robust identifiability [42]. These results provide conditions for strict identifiability of the model, and here, the dimensionality of the latent space is required to be of the same order as the observed space dimensionality. In contrast, a number of recent works analyze *generic* identifiability of overcomplete CP decomposition, which is weaker than strict identifiability, e.g. [103, 116, 151, 68, 53, 45, 54]. These works assume that the factors (i.e. the components) of the CP decomposition are generically drawn and provide conditions for uniqueness. They allow for the latent dimensionality to be much larger (polynomially larger) than the observed dimensionality. These results on the uniqueness of CP decompositions also lead to identifiability of other latent variable models, such as latent tree models, e.g. [7, 6], and the single-topic model, or more generally latent Dirichlet allocation (LDA).

In contrast to the above works dealing with the CP tensor decomposition, we require uniqueness for a more general class of tensor decompositions, in order to establish identifiability of topic models with arbitrarily correlated topics. We establish that our class of tensor decomposition is contained in the class of *Tucker* decompositions which is more general than CP decomposition. Moreover, we explicitly characterize the effect of the sparsity pattern of the factors (i.e., the topic-word matrix) on model identifiability, while all the previous works based on generic identifiability assume fully

dense factors (since sparse factors are not generic). For a general overview of tensor decompositions, see [108, 114].

5.3.0.2 Identifiability and Learning of Undercomplete/Over-determined latent Representations

Much of the theoretical results on identifiability and learning of the latent variable models are limited to non-singular models, which implies that the latent space dimensionality is at most the observed dimensionality. We outline some of the recent works below.

The works of [10, 8, 15] provide an efficient moment-based approach for learning topic models, under constraints on the distribution of the topic proportions, e.g. the single topic model, and more generally latent Dirichlet allocation (LDA). In addition, the approach can handle a variety of latent variable models such as Gaussian mixtures, hidden Markov models (HMM) and community models [13]. The high-level idea is to reduce the problem of learning of the latent variable model to finding a CP decomposition of the (suitably adjusted) observed moment tensor. Various approaches can then be employed to find the CP decomposition. In [15], a tensor power method approach is analyzed and is shown to be an efficient guaranteed recovery method in the non-degenerate (i.e. undercomplete) setting. Previously, simultaneous diagonalization techniques have been employed for solving the CP decomposition, e.g. [10, 129, 52]. However, these techniques fail when the model is overcomplete, as considered here. We note that some recent techniques, e.g. [68], can be employed instead, albeit at a cost of higher computational complexity for overcomplete CP tensor decomposition. However, it is not clear how the sparsity constraints affect the guarantees of such methods. Moreover, these approaches cannot handle general topic models, where the distribution of the topic proportions is not limited to these classes (i.e. either single topic or Dirichlet distribution), and we require tensor decompositions which are more general than the CP decomposition.

There are many other works which consider learning mixture models when multiple views are available. See [10] for a detailed description of these works. Recently, [135] consider learning discrete mixtures given a large number of "views", and they refer to the number of views as the

sampling aperture. They establish improved recovery results (in terms of ℓ_1 bounds) when sufficient number of views are available ($2k-1$ views for a k-component mixture). However, their results are limited to discrete mixtures or single-topic models, while our setting can handle more general topic models. Moreover, our approach is different since we incorporate sparsity constraints in the topic-word distribution. Another series of recent works by [28, 30] employ approaches based on non-negative matrix factorization (NMF) to recover the topic-word matrix. These works allow models with arbitrarily correlated topics, as considered here. They establish guaranteed learning when every topic has an *anchor* word, i.e. the word is uniquely generated from that topic, and does not occur under any other topic. Note that the anchor-word assumption cannot be satisfied in the overcomplete setting.

Our work is closely related to the work of [9] which considers identifiability and learning of topic models under expansion conditions on the topic-word matrix. The work of [149] considers the problem of dictionary learning, which is closely related to the setting of [9], but in addition assumes that the coefficient matrix is random. However, these works in [9, 149] can handle only the undercomplete setting, where the number of topics is less than the dimensionality of the word vocabulary (or the number of dictionary atoms is less than the number of observations in [149]). We extend these results to the overcomplete setting by proposing novel higher order expansion conditions on the topic-word matrix, and also incorporate additional rank constraints present in higher order moments.

5.3.0.3 Dictionary Learning/Sparse Coding

Overcomplete representations have been very popular in the context of dictionary learning or sparse coding. Here, the task is to jointly learn a dictionary as well as a sparse selection of the dictionary atoms to fit the observed data. There have been Bayesian as well as frequentist approaches for dictionary learning [119, 110, 136]. However, the heuristics employed in these works [119, 110, 136] have no performance guarantees. The work of [149] considers learning (undercomplete) dictionaries and provide guaranteed learning under the assumption that the coefficient matrix is random (distributed as Bernoulli-Gaussian variables). Recent works in [124, 122] provide generalization

bounds for predictive sparse coding, where the goal of the learned representation is to obtain good performance on some predictive task. This differs from our framework since we do not consider predictive tasks here, but the task of recovering the underlying latent representation. [89] consider the problem of identifiability of sparse coding and establish that when the dictionary succeeds in reconstructing a certain set of sparse vectors, then there exists a unique sparse coding, up to permutation and scaling. However, our setting here is different, since we do not assume that a sparse set of topics occur in each document.

5.4 Model

We first introduce some notations, and then we provide the persistent topic model.

5.4.1 Notation

The set $\{1, 2, \ldots, n\}$ is denoted by $[n] := \{1, 2, \ldots, n\}$. Given a set $X = \{1, \ldots, p\}$, set $X^{(n)}$ denotes all ordered n-tuples generated from X. The cardinality of a set S is denoted by $|S|$. For any vector u (or matrix U), the support is denoted by $\mathrm{Supp}(u)$, and the ℓ_0 norm is denoted by $\|u\|_0$, which corresponds to the number of non-zero entries of u, i.e., $\|u\|_0 := |\mathrm{Supp}(u)|$. For a vector $u \in \mathbb{R}^q$, $\mathrm{Diag}(u) \in \mathbb{R}^{q \times q}$ is the diagonal matrix with vector u on its diagonal. The column space of a matrix A is denoted by $\mathrm{Col}(A)$. Vector $e_i \in \mathbb{R}^q$ is the i-th basis vector, with the i-th entry equal to 1 and all the others equal to zero. For $A \in \mathbb{R}^{p \times q}$ and $B \in \mathbb{R}^{m \times n}$, the *Kronecker* product[6] $A \otimes B \in \mathbb{R}^{pm \times qn}$ is defined as [78]

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1q}B \\ a_{21}B & a_{22}B & \cdots & a_{2q}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}B & a_{p2}B & \cdots & a_{pq}B \end{bmatrix},$$

[6]Note that in this chapter we use the notation $\otimes$ to denote the Kronecker product, and use notation $\circ$ to denote the outer or tensor product; see (5.15) for the definition. This is different from previous chapters that we use $\otimes$ to denote the outer product.

and for $A = [a_1|a_2|\cdots|a_r] \in \mathbb{R}^{p\times r}$ and $B = [b_1|b_2|\cdots|b_r] \in \mathbb{R}^{m\times r}$, the *Khatri-Rao* product $A \odot B \in \mathbb{R}^{pm\times r}$ is defined as

$$A \odot B = [a_1 \otimes b_1|a_2 \otimes b_2|\cdots|a_r \otimes b_r]\,.$$

5.4.2 Persistent Topic Model

In this section, the *n-persistent topic model* is introduced and this imposes an additional constraint, known as topic persistence on the popular admixture model[43, 134, 132]. The n-persistent topic model reduces to the bag-of-words admixture model when $n = 1$.

An admixture model specifies a q-dimensional vector of topic proportions $h \in \Delta^{q-1} := \{u \in \mathbb{R}^q : u_i \geq 0, \sum_{i=1}^{q} u_i = 1\}$ which generates the observed variables $x_l \in \mathbb{R}^p$ through vectors $a_1, \ldots, a_q \in \mathbb{R}^p$. This collection of vectors $a_i, i \in [q]$, is referred to as the *population structure* or the *topic-word matrix* [132]. For instance, a_i is the conditional distribution of words given topic i. The latent variable h is a q dimensional random vector $h := [h_1, \ldots, h_q]^\top$ known as proportion vector. A prior distribution $P(h)$ over the probability simplex Δ^{q-1} characterizes the prior joint distribution over the latent variables $h_i,\ i \in [q]$. In the topic modeling, this is the prior distribution over the q topics.

The n-persistent topic model has a three-level multi-view hierarchy in Figure 5.1. $2rn$ number of words (views) are shown in the model for some integer $r \geq 1$. In this model, a common hidden topic is persistent for a sequence of n words $\{x_{(j-1)n+1}, \ldots, x_{(j-1)n+n}\}, j \in [2r]$. Note that the random observed variables (words) are exchangeable within groups of size n, where n is the persistence level, but are not globally exchangeable.

We now describe a linear representation of the n-persistent topic model, on lines of [15], but with extensions to incorporate persistence. Each random variable $y_j, j \in [2r]$, is a discrete valued random variable taking one of the q possibilities $\{1, \ldots, q\}$, i.e., $y_j \in [q]$ for $j \in [2r]$. In the n-persistent model, a single common topic is chosen for a sequence of n words $\{x_{(j-1)n+1}, \ldots, x_{(j-1)n+n}\}, j \in [2r]$, i.e., the topic is persistent for n successive views. For notational purposes, we equivalently assume that variables $y_j, j \in [2r]$, are encoded by the basis vectors $e_i,\ i \in [q]$. Thus, the variable

$y_j, j \in [2r]$, is

$$y_j = e_i \in \mathbb{R}^q \iff \text{the topic of the } j\text{-th group of words is } i.$$

Given proportion vector h, topics $y_j, j \in [2r]$, are independently drawn according to the conditional expectation

$$\mathbb{E}\big[y_j|h\big] = h, \quad j \in [2r],$$

or equivalently $\Pr\big[y_j = e_i|h\big] = h_i, j \in [2r], i \in [q]$.

Finally, at the bottom layer, each observed variable x_l for $l \in [2rn]$, is a discrete-valued p-dimensional random variable, where p is the size of word vocabulary. Again, we assume that variables x_l, are encoded by the basis vectors $e_k,\ k \in [p]$, such as

$$x_l = e_k \in \mathbb{R}^p \iff \text{the } l\text{-th word in the document is } k.$$

Given the corresponding topic $y_j, j \in [2r]$, words $x_l, l \in [2rn]$, are independently drawn according to the conditional expectation

$$\mathbb{E}\big[x_{(j-1)n+k}|y_j = e_i\big] = a_i,\ i \in [q], j \in [2r],\ k \in [n], \tag{5.1}$$

where vectors $a_i \in \mathbb{R}^p,\ i \in [q]$, are the conditional probability distribution vectors. The matrix $A = [a_1|a_2|\cdots|a_q] \in \mathbb{R}^{p\times q}$ collecting these vectors is the *population structure* or *topic-word matrix*.

The $(2rn)$-th order moment of observed variables $x_l \in \mathbb{R}^p, l \in [2rn]$, for some integer $r \geq 1$, is defined as (in the matrix form) [7]

$$M_{2rn}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_{rn})(x_{rn+1} \otimes x_{rn+2} \otimes \cdots \otimes x_{2rn})^\top\right] \in \mathbb{R}^{p^{rn}\times p^{rn}}. \tag{5.2}$$

We now briefly remind why this matrix corresponds to the $(2rn)$-th order moment. Let vectors $\mathbf{i} := (i_1, \ldots, i_{rn})$ and $\mathbf{j} := (j_1, \ldots, j_{rn})$ index the rows and columns of moment matrix $M_{2rn}(x)$.

[7] Vector x is the vector generated by concatenating all vectors $x_l, l \in [2rn]$.

Then, from the above definition, the $(\mathbf{i}, \mathbf{j})$-th entry of $M_{2rn}(x)$ is equal to

$$\mathbb{E}\left[(x_1)_{i_1} \cdots (x_{rn})_{i_{rn}} (x_{rn+1})_{j_1} \cdots (x_{2rn})_{j_{rn}}\right],$$

which specifies the corresponding $(2rn)$-th observed moment.

For the n-persistent topic model with $2rn$ number of observations (words) $x_l, l \in [2rn]$, the corresponding moment is denoted by $M_{2rn}^{(n)}(x)$. Note that to estimate the $(2rn)^{\text{th}}$ moment, we require a minimum of $2rn$ words in each document. We can select the first $2rn$ words in each document, and average over the different documents to obtain a consistent estimate of the moment. In this work, we consider the problem of identifiability when exact moments are available.

The moment characterization of the n-persistent topic model is provided in Lemma 2 in Section 5.6.1. Given $M_{2rn}^{(n)}(x)$, what are the sufficient conditions under which the population structure A is identifiable? This is answered in Section 5.5.

Remark 16. Note that our results are valid for the more general linear model $x_l = Ay_j$ (more precisely, $x_{(j-1)n+k} = Ay_j, j \in [2r], k \in [n]$), i.e., each column of matrix A does not need to be a valid probability distribution. Furthermore, the observed random variables x_l, can be continuous while the hidden ones y_j are assumed to be discrete.

5.5 Sufficient Conditions for Generic Identifiability

In this section, the identifiability result for the n-persistent topic model with access to $(2n)$-th order observed moment is provided. First, sufficient deterministic conditions on the population structure A are provided for identifiability in Theorem 5.1. Next, the deterministic analysis is specialized to a random structured model in Theorem 5.2.

We now make the notion of identifiability precise. As defined in literature, (strict) identifiability means that the population structure A can be uniquely recovered up to permutation and scaling for all $A \in \mathbb{R}^{p \times q}$. Instead, we consider a more relaxed notion of identifiability, known as generic identifiability.

Definition 5.1 (Generic identifiability)**.** *We refer to a matrix $A \in \mathbb{R}^{p \times q}$ as generic, with a fixed sparsity pattern when the nonzero entries of A are drawn from a distribution which is absolutely continuous with respect to Lebesgue measure*[8]*. For a given sparsity pattern, the class of population structure matrices is said to be* generically identifiable *[6], if all the non-identifiable matrices form a set of Lebesgue measure zero.*

The $(2r)$-th order moment of hidden variables $h \in \mathbb{R}^q$, denoted by $M_{2r}(h) \in \mathbb{R}^{q^r \times q^r}$, is defined as

$$M_{2r}(h) := \mathbb{E}\left[\left(\overbrace{h \otimes \cdots \otimes h}^{r \text{ terms}}\right)\left(\overbrace{h \otimes \cdots \otimes h}^{r \text{ terms}}\right)^\top\right] \in \mathbb{R}^{q^r \times q^r}. \tag{5.3}$$

We now provide a set of sufficient conditions for generic identifiability of structured topic models given $(2rn)$-th order observed moment. We first start with a natural assumption on the hidden variables.

Condition 1 (Non-degeneracy)**.** *The $(2r)$-th order moment of hidden variables $h \in \mathbb{R}^q$, defined in equation* (5.3)*, is full rank (non-degeneracy of hidden nodes).*

Note that there is no hope of distinguishing distinct hidden nodes without this non-degeneracy assumption. We do not impose any other assumption on hidden variables and can incorporate arbitrarily correlated topics.

Furthermore, we can only hope to identify the population structure A up to scaling and permutation. Therefore, we can identify A up to a canonical form defined as:

Definition 5.2 (Canonical form)**.** *Population structure A is said to be in* canonical form *if all of its columns have unit norm.*

[8] As an equivalent definition, if the non-zero entries of an arbitrary sparse matrix are independently perturbed with noise drawn from a continuous distribution to generate A, then A is called generic.

5.5.1 Deterministic Conditions for Generic Identifiability

In this section, we consider a fixed sparsity pattern on the population structure A and establish generic identifiability when non-zero entries of A are drawn from some continuous distribution. Before providing the main result, a generalized notion of (perfect) matching for bipartite graphs is defined. We subsequently impose these conditions on the bipartite graph from topics to words which encodes the sparsity pattern of population structure A.

5.5.1.1 Generalized Matching for Bipartite Graphs

A bipartite graph with two disjoint vertex sets Y and X and an edge set E between them is denoted by $G(Y, X; E)$. Given the bi-adjacency matrix A, the notation $G(Y, X; A)$ is also used to denote a bipartite graph. Here, the rows and columns of matrix $A \in \mathbb{R}^{|X|\times|Y|}$ are respectively indexed by X and Y vertex sets. For any subset $S \subseteq Y$, the set of neighbors of vertices in S with respect to A is defined as $N_A(S) := \{i \in X : A_{ij} \neq 0 \text{ for some } j \in S\}$, or equivalently, $N_E(S) := \{i \in X : (j, i) \in E \text{ for some } j \in S\}$ with respect to edge set E.

Here, we define a generalized notion of matching for a bipartite graph and refer to it as n-gram matching.

Definition 5.3 ((Perfect) n-gram matching)**.** *A* n-gram matching M *for a bipartite graph* $G(Y, X; E)$ *is a subset of edges* $M \subseteq E$ *which satisfies the following conditions. First, for any* $j \in Y$*, we have* $|N_M(j)| \leq n$*. Second, for any* $j_1, j_2 \in Y, j_1 \neq j_2$*, we have* $\min\{|N_M(j_1)|, |N_M(j_2)|\} > |N_M(j_1) \cap N_M(j_2)|$.

A perfect n-gram matching *or* Y-saturating n-gram matching *for the bipartite graph* $G(Y, X; E)$ *is a* n*-gram matching* M *in which each vertex in* Y *is* exactly *connected to* n *edges in* M.

In words, in a n-gram matching M, each vertex $j \in Y$ is at most connected to n edges in M and for any pair of vertices in Y ($j_1, j_2 \in Y, j_1 \neq j_2$), there exists at least one non-common neighbor in set X for each of them (j_1 and j_2).

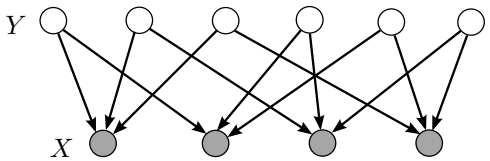


Figure 5.2: A bipartite graph $G(Y, X; E)$ with $|X| = 4$ and $|Y| = 6$ where the edge set E itself is a perfect 2-gram matching.

As an example, a bipartite graph $G(Y, X; E)$ with $|X| = 4$ and $|Y| = 6$ is shown in Figure 5.2 for which the edge set E itself is a perfect 2-gram matching.

We also define the following definition of a n-gram matrix.

Definition 5.4 (n-gram Matrix)**.** *Given a matrix $A \in \mathbb{R}^{p\times q}$, its n-gram matrix $A^{\odot n} \in \mathbb{R}^{p^n\times q}$ is defined as the matrix whose $(\mathbf{i}, j)$-th entry is given by, for $\mathbf{i} := (i_1, i_2, \ldots, i_n) \in [p]^n$ and $j \in [q]$,*

$$A^{\odot n}(\mathbf{i}, j) := A_{i_1,j}A_{i_2,j}\cdots A_{i_n,j}, \quad \text{or} \quad A^{\odot n} := \overbrace{A \odot \cdots \odot A}^{n \text{ times}}.$$

That is, $A^{\odot n}$ is the column-wise n^{th} order Kronecker product of n copies of A, and is known as the Khatri-Rao product [78]. Given bipartite graph $G(Y, X; A)$, the notation $G(Y, X^{(n)}; A^{\odot n})$ is also used to denote the bipartite graph corresponding to bi-adjacency matrix $A^{\odot n}$. Here $X^{(n)}$ denotes all ordered n-tuples generated from elements of set X which indexes the rows of $A^{\odot n}$.

The above two definitions might seem unrelated at the first glance, but the following lemma connects them where an interesting property is stated relating the existence of perfect matching in $G(Y, X^{(n)}; A^{\odot n})$ to the existence of perfect n-gram matching in $G(Y, X; A)$. This property is also the original motivation behind defining such notion of generalized matching.

Lemma 1. *If $G(Y, X; A)$ has a perfect n-gram matching, then $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching. In the other direction, if $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching $M^{\odot n}$, then $G(Y, X; A)$ has a perfect n-gram matching under the following condition on $M^{\odot n}$. All the matching edges $(j, (i_1, \ldots, i_n)) \in M^{\odot n}$ should satisfy $i_1 \neq i_2 \neq \cdots \neq i_n$ for all $j \in Y$. In words, the matching edges should be connected to nodes in $X^{(n)}$, which are indexed by tuples of distinct indices.*

See Appendix E.1.4 for the proof.

We also provide more discussions and remarks on the n-gram matching as follows.

Remark 17 (Relationship to other matchings). The relationship of n-gram matching to other types of matchings is discussed below.

- Regular matching: For special case $n = 1$, the (perfect) n-gram matching reduces to the usual (perfect) matching for bipartite graphs.
- b-matching: For a bipartite graph $G(Y, X; E)$, a b-matching for vertices in Y is a subset of edges $M_b \subseteq E$, where each vertex in Y is connected to b edges. Comparing with the proposed perfect (Y-saturating) b-gram matching, b-matching does not enforce that the set of neighbors be different.

Remark 18 (Necessary size bound). Consider a bipartite graph $G(Y, X; E)$ with $|Y| = q$ and $|X| = p$ which has a perfect n-gram matching. Note that there are $\binom{p}{n}$ n-combinations on X side and each combination can at most have one neighbor (a node in Y which is connected to all nodes in the combination) through the matching, and therefore we necessarily have $q \leq \binom{p}{n}$.

Finally, note that the existence of perfect n-gram matching results in the existence of perfect $(n+1)$-gram matching [9], but the reverse is not true. For example, the bipartite graph $G(Y, X; E)$ with $|X| = 4$ and $|Y| = \binom{4}{2} = 6$ in Figure 5.2, has a perfect 2-gram matching, but not a perfect (1-gram) matching (since $6 > 4$).

5.5.1.2 Identifiability Conditions Based on Existence of Perfect n-gram Matching in Topic-word Graph

Now, we are ready to propose the identifiability conditions and result.

Condition 2 (Perfect n-gram matching on A)**.** *The bipartite graph $G(V_h, V_o; A)$ between hidden and observed variables, has a perfect n-gram matching* [10].

[9] Note that the degree of each node (on matching side Y) in the original bipartite graph should be at least $n + 1$.

[10] Parameter n in all of the conditions refer to the same parameter n as the persistence level of the model. Note that we are considering the n-persistent topic model proposed in Section 5.4.

The above condition implies that the sparsity pattern of matrix A is appropriately scattered in the mapping from hidden to observed variables to be identifiable. Intuitively, it means that every hidden node can be distinguished from another hidden node by its unique set of neighbors under the corresponding n-gram matching.

Furthermore, condition 2 is the key to be able to propose identifiability in the overcomplete regime. As stated in the size bound in Remark 18, for $n \geq 2$, the number of hidden variables can be more than the number of observed variables and we can still have perfect n-gram matching.

Definition 5.5 (Kruskal rank, [112])**.** *The* Kruskal rank *or the* krank *of matrix A is defined as the maximum number k such that every subset of k columns of A is linearly independent.*

Note that krank is different from the general notion of matrix rank and it is a lower bound for the matrix rank, i.e., $\text{Rank}(A) \geq \text{krank}(A)$.

Condition 3 (Krank condition on A)**.** *The Kruskal rank of matrix A satisfies the bound* $\text{krank}(A) \geq d_{\max}(A)^n$*, where $d_{\max}(A)$ is the maximum node degree of any column of A, i.e., $d_{\max}(A) := \max_{i \in [q]} \|Ae_i\|_0$. Here n is the same as parameter n in Condition 2.*

In the overcomplete regime, it is not possible for A to be full column rank and $\text{krank}(A) < |V_h| = q$. However, note that a large enough krank ensures that appropriate sized subsets of columns of A are linearly independent. For instance, when $\text{krank}(A) > 1$, any two columns cannot be collinear and the above condition rules out the collinear case for identifiability. In the above condition, we see that a larger krank can incorporate denser connections between topics and words.

On the other hand, the bound in Condition 3 imposes sparsity on the columns of topic-word matrix as $d_{\max}(A) \leq \text{krank}(A)^{1/n}$. Under such sparsity constraint, each topic (indexing the columns of A) is supported on a specific set of words which enables us to distinguish between different topics and identify the model. But, it seems that this bound is not tight[11].

[11] The looseness originates from bound (E.13) as $\left|N_{A^{\odot n}_{\text{Rest.}}}(S)\right| \geq |N_A(S)| + |S|$ in the proof. See Definitions 5.4 and E.1 for the definition of $A^{\odot n}_{\text{Rest.}}$. Note that many terms in this lower bound on $\left|N_{A^{\odot n}_{\text{Rest.}}}(S)\right|$ are ignored which leads to a loose bound that might be improved.

The main identifiability result under a fixed graph structure is stated in the following theorem for $n \geq 2$, where n is the topic persistence level. The identifiability result relies on having access to the $(2rn)$-th order moment of observed variables $x_l, l \in [2rn]$, defined in equation (5.2) as

$$M_{2rn}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_{rn})(x_{rn+1} \otimes x_{rn+2} \otimes \cdots \otimes x_{2rn})^\top\right] \in \mathbb{R}^{p^{rn} \times p^{rn}},$$

for some integer $r \geq 1$.

Theorem 5.1 (Generic identifiability under deterministic topic-word graph structure)**.** *Let $M_{2rn}^{(n)}(x)$ in equation* (5.2) *be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model satisfies conditions 1, 2 and 3, then, for any $n \geq 2$, all the columns of population structure A are* generically identifiable *from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of the hidden variables, denoted by $M_{2r}(h)$, is also* generically identifiable.

The theorem is proved in Appendix E.1. It is seen that the population structure A is identifiable, given any observed moment of order at least $2n$. Increasing the order of observed moment results in identifying higher order moments of the hidden variables.

The above theorem does not cover the case when the persistence level $n = 1$. This is the usual bag-of-words admixture model. Identifiability of this model has been studied earlier in [9] and we recall it below.

Remark 19 (Bag-of-words admixture model, [9]). Given $(2r)$-th order observed moments with $r \geq 1$, the structure of the popular bag-of-words admixture model and the $(2r)$-th order moment of hidden variables are identifiable, when A is full column rank and the following expansion condition holds [9]

$$|N_A(S)| \geq |S| + d_{\max}(A), \quad \forall S \subseteq V_h,\ |S| \geq 2. \tag{5.4}$$

Our result for $n \geq 2$ in Theorem 5.1, provides identifiability in the overcomplete regime with weaker matching condition 2 and krank condition 3. The matching condition 2 is weaker than the above expansion condition which is based on the perfect matching and hence, does not allow overcomplete

models. Furthermore, the above result for the bag-of-words admixture model requires full column rank of A which is more stringent than our krank condition 3.

Remark 20 (Kruskal rank and degree diversity). Condition 3 requires that the Kruskal rank of the topic-word matrix be large enough compared to the maximum degree of the topics. Intuitively, a larger Kruskal rank ensures enough diversity in the word supports among different topics under a higher level of sparsity. This Kruskal rank condition also allows for more degree diversity among the topics, when the topic persistence level $n > 1$. On the other hand, for the bag-of-words model ($n = 1$), using (5.4) implies that $2d_{\min} > d_{\max}$, where $d_{\min}, d_{\max}$ are the minimum and maximum degrees of the topics. Thus, we provide identifiability results with more degree diversity when higher order moments are employed.

Remark 21 (Recovery using ℓ_1 optimization). It turns out that our conditions for identifiability imply that the columns of the n-gram matrix $A^{\odot n}$, defined in Definition 5.4, are the sparsest vectors in $\mathrm{Col}\Big(M_{2n}^{(n)}(x)\Big)$, having a tensor rank of one. See Appendix E.1. This implies recovery of the columns of A through exhaustive search, which is not efficient. On the other hand, efficient ℓ_1-based recovery algorithms have been analyzed in [148, 9] for the undercomplete case ($n = 1$). They can be employed here for recovery from higher order moments as well. Exploiting additional structure present in $A^{\odot n}$, for $n > 1$, such as rank-1 test devices proposed in [68] are interesting avenues for future investigation.

In Theorem 5.1, we provide our identifiability result for the overcomplete topic-word matrix A under topic persistent model. The result for the bag-of-words admixture model is also reviewed in Remark 19 under the assumption that A is full column rank. In the following corollary, we provide the strong identifiability result for the full column rank topic-word matrix under the topic persistent model.

Corollary 5.1 (Identifiability for undercomplete topic-word matrix)**.** *Let $M_{2rn}^{(n)}(x)$ in equation* (5.2) *be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model satisfies condition 1, and in addition A is full column rank, then for any $n \geq 2$, all the columns of population structure A are* generically identifiable *from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of the hidden variables, denoted by $M_{2r}(h)$, is also* generically identifiable.

Comparing to Theorem 5.1 and Remark 19, the expansion (and krank) conditions are not required in the above result which is a huge relaxation. The reason is both undercomplete regime and topic persistence are assumed here which relaxes the other conditions. Note that the assumptions that topic persists with persistence $n \geq 2$, and the topic-word matrix is full column rank (and therefore undercomplete) is reasonable in many applications.

5.5.2 Analysis Under Random Topic-word Graph Structures

In this section, we specialize the identifiability result to the random case. This result is based on more transparent conditions on the size and the degree of the random bipartite graph $G(V_h, V_o; A)$. We consider the random model where in the bipartite graph $G(V_h, V_o; A)$, each node $i \in V_h$ is randomly connected to d_i different nodes in set V_o. Note that this is a heterogeneous degree model.

Furthermore, the random identifiability result is provided with high probability which is defined as follows.

Definition 5.6 (**whp**). *A sequence of events $\mathcal{E}_p$ (depending on size parameter p) occurs* with high probability *(**whp**) if $\Pr(\mathcal{E}_p) = 1 - O(p^{-\epsilon})$ for some $\epsilon > 0$.*

Condition 4 (Size condition). *The random bipartite graph $G(V_h, V_o; A)$ with $|V_h| = q, |V_o| = p$, and $A \in \mathbb{R}^{p \times q}$, satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ for some constant $0 < c < 1$.*

This size condition is required to establish that the random bipartite graph has a perfect n-gram matching (and hence satisfies deterministic condition 2). It is shown in Section 5.7.2.1 that the necessary size constraint $q = O(p^n)$ stated in Remark 18, is achieved in the random case. Thus, the above constraint allows for the overcomplete regime, where $q \gg p$ for $n \geq 2$, and is tight.

Condition 5 (Degree condition). *In the random bipartite graph $G(V_h, V_o; A)$ with $|V_h| = q, |V_o| = p$, and $A \in \mathbb{R}^{p \times q}$, the degree d_i of nodes $i \in V_h$ satisfies the following lower and upper bounds ($d_i \in [d_{\min}, d_{\max}]$):*

- Lower bound: $d_{\min} \geq \max\{1+\beta \log p, \alpha \log p\}$ *for some constants* $\beta > \frac{n-1}{\log 1/c}, \alpha > \max\{2n^2\big(\beta \log \frac{1}{c} + 1\big), 2\beta n\}$.

Parameter	Representing
p	dimension of observed variables
q	dimension of hidden variables
n	persistence level
c	size ratio such that $q \leq \left(c\frac{p}{n}\right)^n$
α, β	Constants for lower bound on degree such that $d_{\min} \geq \max\{1 + \beta \log p, \alpha \log p\}$

Table 5.1: Table of parameters.

- Upper bound: $d_{\max} \leq (cp)^{\frac{1}{n}}$.

Intuitively, the lower bound on the degree is required to show that the corresponding bipartite graph $G(V_h, V_o; A)$ has sufficient number of random edges to ensure that it has perfect n-gram matching with high probability. The upper bound on the degree is mainly required to satisfy the krank condition 3, where $d_{\max}(A)^n \leq \text{krank}(A)$. As discussed after Condition 3, this upper bound is not tight.

It is important to see that, for $n \geq 2$, the above condition on degree covers a range of models from sparse to intermediate regimes and it is reasonable in a number of applications that each topic does not generate a very large number of words.

The proposed parameters in Conditions 4 and 5 are summarized in Table 5.1.

The main random identifiability result is stated in the following theorem for $n \geq 2$, while $n = 1$ case is addressed in Remark 23. The identifiability result relies on having access to the $(2rn)$-th order moment of observed variables $x_l, l \in [2rn]$, defined in equation (5.2) as

$$M_{2rn}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_{rn})(x_{rn+1} \otimes x_{rn+2} \otimes \cdots \otimes x_{2rn})^\top\right] \in \mathbb{R}^{p^{rn} \times p^{rn}},$$

for some integer $r \geq 1$.

Probability rate constants: The probability rate of success in the following random identifiability result is specified by constants $\beta' > 0$ and $\gamma = \gamma_1 + \gamma_2 > 0$ as

$$\beta' = -\beta \log c - n + 1, \tag{5.5}$$

$$\gamma_1 = e^{n-1}\Big(\frac{c}{n^{n-1}} + \frac{e^2}{1-\delta_1} n^{\beta'+1}\Big), \tag{5.6}$$

$$\gamma_2 = \frac{c^{n-1}e^2}{n^n(1-\delta_2)}, \tag{5.7}$$

where δ_1 and δ_2 are some constants satisfying $e^2\Big(\frac{p}{n}\Big)^{-\beta \log 1/c} < \delta_1 < 1$ and $\frac{c^{n-1}e^2}{n^n} p^{-\beta'} < \delta_2 < 1$.

Theorem 5.2 (Random identifiability)**.** *Let $M_{2rn}^{(n)}(x)$ in equation (5.2) be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model with random population structure A satisfies conditions 1, 4 and 5, then **whp** (with probability at least $1 - \gamma p^{-\beta'}$ for constants $\beta' > 0$ and $\gamma > 0$, specified in (5.5)-(5.7)), for any $n \geq 2$, all the columns of population structure A are identifiable from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of hidden variables, denoted by $M_{2r}(h)$, is also identifiable, **whp**.*

The theorem is proved in Appendix E.2. Similar to the deterministic analysis, it is seen that the population structure A is identifiable given any observed moment with order at least $2n$. Increasing the order of observed moment results in identifying higher order moments of the hidden variables.

Remark 22 (Trade-off between topic-word size ratio and degree). When the number of hidden variables increases, i.e. c increases, but the order n is kept fixed, the bounds on degree in condition 5 also needs to grow. Intuitively, a larger degree is needed to provide more flexibility in choosing the subsets of neighbors for hidden nodes to ensure the existence of a perfect n-gram matching in the bipartite graph, which in turn ensures identifiability. Note that as c grows, the parameter β, which is the lower bound on d also grows, and the probability rate (i.e., the term $-\beta \log c$) remains constant. Hence, the probability rate does not change as c increases, since the increase in the degree d compensates the additional "difficulty" arising due to a larger number of hidden variables.

The above identifiability theorem only covers for $n \geq 2$ and the $n = 1$ case is addressed in the following remark.

Remark 23 (Bag-of-words admixture model). The identifiability result for the random bag-of-words admixture model is comparable to the result in [148], which considers exact recovery of sparsely-used dictionaries. They assume that $Y = DX$ is given for some unknown arbitrary dictionary $D \in \mathbb{R}^{q\times q}$ and unknown random sparse coefficient matrix $X \in \mathbb{R}^{q\times p}$. They establish that if $D \in \mathbb{R}^{q\times q}$ is full rank and the random sparse coefficient matrix $X \in \mathbb{R}^{q\times p}$ follows the Bernoulli-subgaussian model with size constraint $p > Cq\log q$ and degree constraint $O(\log q) < \mathbb{E}[d] < O(q\log q)$, then the model is identifiable, whp. Comparing the size and degree constraints, our identifiability result for $n \geq 2$ requires more stringent upper bound on the degree ($d = O(p^{1/n})$), while more relaxed condition on the size ($q = O(p^n)$) which allows to identifiability in the overcomplete regime.

Remark 24 (The size condition is tight). The size bound $q = O(p^n)$ in the above theorem achieves the necessary condition that $q \leq \binom{p}{n} = O(p^n)$ (see Remark 18), and is therefore tight. The sufficiency is argued in Theorem 5.3, where we show that the matching condition 2 holds under the above size and degree conditions 4 and 5.

As in the deterministic case, we finish this section by providing random identifiability result for the full column rank topic-word matrix under the topic persistent model.

Corollary 5.2 (Random identifiability for undercomplete topic-word matrix)**.** *Let $M_{2rn}^{(n)}(x)$ in equation* (5.2) *be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model with random population structure $A \in \mathbb{R}^{p\times q}$ satisfies condition 1, size condition $q \leq cp$ for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq 1 + \beta\log p$ for some constant $\beta > 0$, then **whp** (with probability at least $1 - O(z^{-\beta\log 1/c})$ where $\beta\log\frac{1}{c} > 0$), for any $n \geq 2$, all the columns of population structure A are identifiable from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of hidden variables, denoted by $M_{2r}(h)$, is also identifiable, **whp**.*

Comparing to Theorem 5.2, the upper bound on the degree (sparsity constraint) is not required in the above result which is a huge relaxation.

5.6 Identifiability via Uniqueness of Tensor Decompositions

In this section, we characterize the moments of the n-persistent topic model in terms of the model parameters, i.e. the topic-word matrix A and the moment of hidden variables. We relate identifiability of the topic model to uniqueness of a certain class of tensor decompositions, which in turn, enables us to prove Theorems 5.1 and 5.2. We then discuss the special cases of the persistent topic model, viz., the single topic model (infinite-persistent topic model) and the bag-of-words admixture model (1-persistent topic model).

5.6.1 Moment Characterization of the Persistent Topic Model

In the following lemma, which is proved in Appendix E.1.2, we characterize the observed moments of a persistent topic model. Throughout this section, the order of the observed moment is fixed to $2m$.

Lemma 2 (n-persistent topic model moment characterization)**.** *The $(2m)$-th order moment of observed variables, defined in equation (5.2), for the n-persistent topic model is characterized as*[12]*:*

- *if $m = rn$, for some integer $r \geq 1$, then*

$$M_{2m}^{(n)}(x) = \left(\overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r \text{ times}}\right) M_{2r}(h) \left(\overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r \text{ times}}\right)^{\top}, \tag{5.8}$$

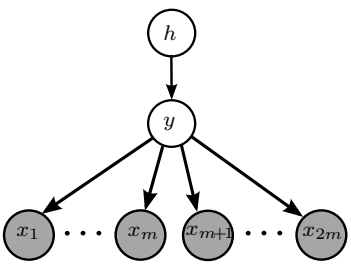
  where $M_{2r}(h) \in \mathbb{R}^{q^r \times q^r}$ is the $(2r)$-th order moment of hidden variables $h \in \mathbb{R}^q$, defined in equation (5.3), and the n-gram matrix $A^{\odot n}$ is defined in Definition 5.4.

- *If $n \geq 2m$, then*

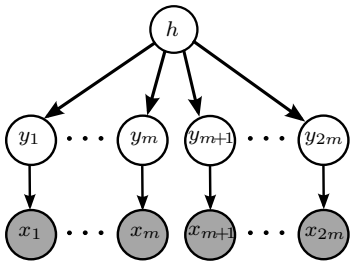
$$M_{2m}^{(n)}(x) = \left(A^{\odot m}\right) M_1(h) \left(A^{\odot m}\right)^{\top}, \tag{5.9}$$

  where $M_1(h) := \mathrm{Diag}(\mathbb{E}[h]) \in \mathbb{R}^{q \times q}$ is the first order moment of hidden variables $h \in \mathbb{R}^q$, stacked in a diagonal matrix.

[12] The other cases not covered in Lemma 2 are deferred to Appendix E.1.2. See Remark 32.



(a) Single topic model
(infinite-persistent topic model)



(b) Bag-of-words admixture model
(1-persistent topic model)

Figure 5.3: Hierarchical structure of the single topic model and bag-of-words admixture model shown for $2m$ number of words (views).

Thus, we see that the observed moments can be expressed in terms of the hidden moments $M(h)$ and the Kronecker products of the n-gram matrices. In the special case, when the persistence level is large enough compared to the order of the moment ($n \geq 2m$), the moment form reduces to a Khatri-Rao product form in (5.9). Moreover, in (5.9), we have a diagonal matrix $M_1(h)$ instead of a general (dense) matrix $M_{2r}(h)$ in (5.8), when $n < 2m = 2rn$. Thus, we have a more succinct representation of the moments in (5.9) when the persistence level of the topics is large enough.

In the following, we contrast the special cases when the persistence level n is $n \to \infty$ (single topic model) and $n = 1$ (bag of words admixture model), as shown in Fig.5.3a and Fig.5.3b. In order to have a fair comparison, the number of observed variables is fixed to $2m$ and the persistence level is varied.

Single topic model ($n \to \infty$): The condition in (5.9) ($n \geq 2m$) is always satisfied for the single-topic model, since $n \to \infty$ in this case, and we have

$$M_{2m}^{(\infty)}(x) = \left(A^{\odot m}\right) M_1(h) \left(A^{\odot m}\right)^\top . \tag{5.10}$$

Note that $M_1(h)$ is a diagonal matrix.

Bag-of-words admixture model ($n = 1$): From Lemma 2, the $(2m)$-th order moment of observed variables $x_l, l \in [2m]$, for the bag-of-words admixture model (1-persistent topic model), shown in

Figure 5.3b, is given by

$$M_{2m}^{(1)}(x) = \Big(\overbrace{A \otimes \cdots \otimes A}^{m \text{ times}}\Big) M_{2m}(h) \Big(\overbrace{A \otimes \cdots \otimes A}^{m \text{ times}}\Big)^\top, \tag{5.11}$$

where $M_{2m}(h) \in \mathbb{R}^{q^m \times q^m}$ is the $(2m)$-th order moment of hidden variables $h \in \mathbb{R}^q$, defined in (5.3). Note that $M_{2m}(h)$ is a full matrix in general.

Contrasting single topic $(n \to \infty)$ *and bag of words models* $(n = 1)$*:* Comparing equations (5.10) and (5.11), it is seen that the moments under the single topic model in (5.10) are more "structured" compared to the bag of words model in (5.11). In (5.11), we have Kronecker products of the topic-word matrix A, while (5.10) involves Khatri-Rao products of A. This forms a crucial criterion in determining of whether overcomplete models are identifiable, as discussed below.

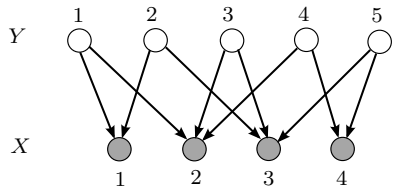
Why does persistence help in identifiability of overcomplete models? For simplicity, let the order of the moment $2m = 4$. The equations (5.10) and (5.11) reduce to

$$M_4^{(\infty)}(x) = (A \odot A)\, \mathrm{Diag}\left(\mathbb{E}\big[h\big]\right) (A \odot A)^\top, \tag{5.12}$$

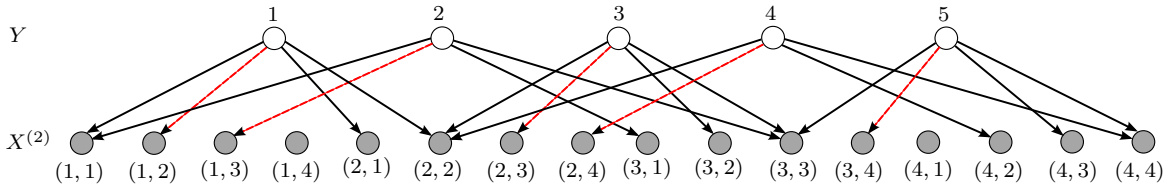
$$M_4^{(1)}(x) = (A \otimes A) \mathbb{E}\big[(h \otimes h)(h \otimes h)^\top\big] (A \otimes A)^\top. \tag{5.13}$$

Note that for the single topic model in (5.12), the Khatri-Rao product matrix $A \odot A \in \mathbb{R}^{p^2 \times q}$ has the same as the number of columns (i.e. the latent dimensionality) of the original matrix A, while the number of rows (i.e. the observed dimensionality) is increased. Thus, the Khatri-Rao product "expands" the effect of hidden variables to higher order observed variables, which is the key towards identifying overcomplete models. In other words, the original overcomplete representation becomes determined due to the 'expansion effect' of the Khatri-Rao product structure of the higher order observed moments.

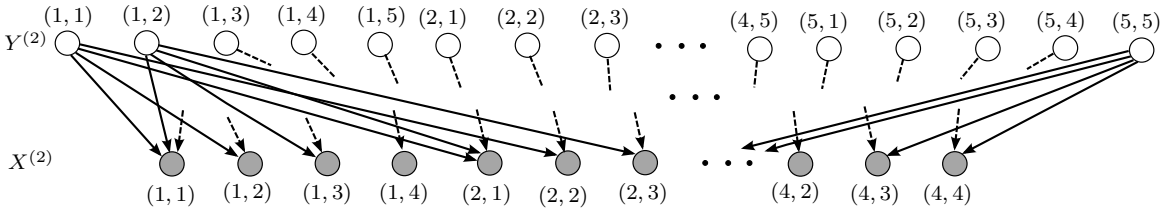
On the other hand, in the bag-of-words admixture model in (5.13), this interesting 'expansion property' does not occur, and we have the Kronecker product $A \otimes A \in \mathbb{R}^{p^2 \times q^2}$, in place of the Khatri-Rao products. The Kronecker product operation increases both the number of the columns



(a) Structure of an overcomplete matrix $A \in \mathbb{R}^{4\times 5}$ having a perfect 2-gram matching.



(b) Structure of $A \odot A \in \mathbb{R}^{16\times 5}$ having a perfect (Y-saturating) matching, highlighted by dashed red edges.



(c) Structure of $A \otimes A \in \mathbb{R}^{16\times 25}$. For simplicity, only a few edges and nodes are shown and the dashed edges denote the bunch of edges connected to each node, not specifically shown.

Figure 5.4: An example of an overcomplete matrix A and the matrices $A \odot A$ and $A \otimes A$. The corresponding bipartite graphs encode the sparsity pattern of each of the matrices. $A \odot A$ expands the effect of hidden variables to second order observed variables which is crucial for overcomplete identifiability, while in the $A \otimes A$, the order of both the hidden and observed variables are increased.

(i.e. latent dimensionality) and the number of rows (i.e. observed dimensionality), which implies that higher order moments do not help in identifying overcomplete models.

An example is provided in Figure 5.4 which helps to see how the matrices $A \odot A$ and $A \otimes A$ behave differently in terms of mapping topics to word tuples.

Note that for the n-persistent model, for $n = 2$, the 4^{th} order moment reduces to

$$M_4^{(2)}(x) = (A \odot A)\mathbb{E}\big[hh^\top\big](A \odot A)^\top. \tag{5.14}$$

Contrasting the above equation with (5.12) and (5.13), we find that the 2-persistent model retains the desirable property of possessing Khatri-Rao products, while being more general than the form

for single topic model in (5.12). This key property enables us to establish identifiability of topic models with finite persistence levels.

5.6.2 Tensor Algebra of the Moments

In Section 5.6.1, we provided a representation of the moment forms in the matrix form. We now provide the equivalent tensor representation of the moments. The tensor representation is more compact and transparent, and allows us to compare the topic models under different levels of persistence. We compare the derived tensor form with the well-known Tucker and CP decompositions. We first introduce some tensor notations and definitions.

5.6.2.1 Tensor Notations and Definitions

A real-valued order-n tensor $A \in \bigotimes_{i=1}^{n} \mathbb{R}^{p_i} := \mathbb{R}^{p_1 \times \cdots \times p_n}$ is a n dimensional array $A(1:p_1, \ldots, 1:p_n)$, where the i-th mode is indexed from 1 to p_i. In this work, we restrict ourselves to the case that $p_1 = \cdots = p_n = p$, and simply write $A \in \bigotimes^n \mathbb{R}^p$. A *fiber* of a tensor A is a vector obtained by fixing all indices of A except one, e.g., for $A \in \bigotimes^4 \mathbb{R}^3$, the vector $f = A(2, 1:3, 3, 1)$ is a fiber.
For a vector $u \in \mathbb{R}^p$, $\text{Diag}_n(u) \in \bigotimes^n \mathbb{R}^p$ is the n-th order diagonal tensor with vector u on its diagonal. The tensor $A \in \bigotimes^n \mathbb{R}^p$, is stacked as a vector $a \in \mathbb{R}^{p^n}$ by the vec$(\cdot)$ operator, defined as

$$a = \text{vec}(A) \Leftrightarrow a\big((i_1 - 1)p^{n-1} + (i_2 - 1)p^{n-2} + \cdots + (i_{n-1} - 1)p + i_n\big) = A(i_1, i_2, \ldots, i_n).$$

The inverse of $a = \text{vec}(A)$ operation is denoted by $A = \text{ten}(a)$.
For vectors $a_i \in \mathbb{R}^{p_i}, i \in [n]$, the tensor outer product operator "$\circ$" is defined as [78]

$$A = a_1 \circ a_2 \circ \cdots \circ a_n \in \bigotimes_{i=1}^{n} \mathbb{R}^{p_i} \Leftrightarrow A(i_1, i_2, \ldots, i_n) := a_1(i_1)a_2(i_2)\cdots a_n(i_n). \tag{5.15}$$

The above generated tensor is a rank-1 tensor. The *tensor rank* is the minimal number of rank-1 tensors into which a tensor can be decomposed. This type of rank is called CP (Candecomp/Parafac) tensor rank in the literature [78].

According to above definitions, for any set of vectors $a_i \in \mathbb{R}^{p_i}, i \in [n]$, we have the following pair of equalities:

$$\mathrm{vec}(a_1 \circ a_2 \circ \cdots \circ a_n) = a_1 \otimes a_2 \otimes \cdots \otimes a_n,$$
$$\mathrm{ten}(a_1 \otimes a_2 \otimes \cdots \otimes a_n) = a_1 \circ a_2 \circ \cdots \circ a_n.$$

For any vector $a \in \mathbb{R}^p$, the power notations are also defined as

$$a^{\otimes n} := \overbrace{a \otimes a \otimes \cdots \otimes a}^{n \text{ times}} \in \mathbb{R}^{p^n},$$
$$a^{\circ n} := \overbrace{a \circ a \circ \cdots \circ a}^{n \text{ times}} \in \bigotimes^n \mathbb{R}^p.$$

The second power is usually called the n-th order *tensor power* of vector a.

Finally, the Tucker and CP (Candecomp/Parafac) representations are defined as follows [78, 108].

Definition 5.7 (Tucker representation)**.** *Given a core tensor $S \in \bigotimes_{i=1}^n \mathbb{R}^{r_i}$ and inverse factors $U_i \in \mathbb{R}^{p_i \times r_i}, i \in [n]$, the Tucker representation of the n-th order tensor $A \in \bigotimes_{i=1}^n \mathbb{R}^{p_i}$ is*

$$A = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_n=1}^{r_n} S(i_1, i_2, \ldots, i_n) U_1(:, i_1) \circ U_2(:, i_2) \circ \cdots \circ U_n(:, i_n) =: [[S; U_1, U_2, \ldots, U_n]], \tag{5.16}$$

where $U_j(:, i_j)$ denotes the i_j-th column of matrix U_j. The tensor S is referred to as the core tensor.

Definition 5.8 (CP representation)**.** *Given $\lambda \in \mathbb{R}^r, U_i \in \mathbb{R}^{p_i \times r}, i \in [n]$, the CP representation of the n-th order tensor $A \in \bigotimes_{i=1}^n \mathbb{R}^{p_i}$ is*

$$A = \sum_{i=1}^{r} \lambda_i U_1(:, i) \circ U_2(:, i) \circ \cdots \circ U_n(:, i) =: [[\mathrm{Diag}_n(\lambda); U_1, U_2, \ldots, U_n]], \tag{5.17}$$

where $U_j(:, i)$ denotes the i-th column of matrix U_j.

Note that the CP representation is a special case of the Tucker representation when the core tensor S is square and diagonal.

5.6.2.2 Tensor Representation of Moments Under Topic Model

We now provide a tensor representation of the moments.

For the n-persistent topic model, the $2m$-th observed moment is denoted by $T_{2m}^{(n)}(x)$, which is the tensor form of the moment matrix $M_{2m}^{(n)}(x)$, characterized in Lemma 2. It is given by

$$T_{2m}(x)_{(i_1,i_2,\ldots,i_{2m})} := \mathbb{E}[x_1(i_1)x_2(i_2)\cdots x_{2m}(i_{2m})], \quad i_1, i_2, \ldots, i_{2m} \in [p], \tag{5.18}$$

where $T_{2m}(x) \in \bigotimes^{2m} \mathbb{R}^p$.

This tensor is characterized in the following lemma, and is proved in Appendix E.1.2.

Lemma 3 (n-persistent topic model moment characterization in tensor form)**.** *The $(2m)$-th order moment of words, defined in equation* (5.18)*, for the n-persistent topic model is characterized as*[13]*:*

- *if $m = rn$ for some integer $r \geq 1$, then*

$$\begin{aligned} T_{2m}^{(n)}(x) &= \sum_{i_1=1}^{q} \sum_{i_2=1}^{q} \cdots \sum_{i_{2r}=1}^{q} \mathbb{E}[h_{i_1} h_{i_2} \cdots h_{i_{2r}}] a_{i_1}^{\circ n} \circ a_{i_2}^{\circ n} \circ \cdots \circ a_{i_{2r}}^{\circ n} \\ &= \Big[\Big[S_r; \overbrace{A, A, \ldots, A}^{2m \text{ times}}\Big]\Big], \end{aligned} \tag{5.19}$$

  where $S_r \in \bigotimes^{2rn} \mathbb{R}^q$ is the core tensor in the above Tucker representation with the sparsity pattern as

$$S_r(\mathbf{i}) = \begin{cases} M_{2r}(h)_{\left((i_n, i_{2n}, \ldots, i_{rn}), (i_{(r+1)n}, i_{(r+2)n}, \ldots, i_{2rn})\right)} & , i_1 = i_2 = \cdots = i_n, i_{n+1} = i_{n+2} = \cdots = i_{2n}, \ldots \\ 0 & , \text{o. w.}, \end{cases}$$

  where $\mathbf{i} := (i_1, i_2, \ldots, i_{2rn})$.

[13] The other cases not covered in Lemma 3 are deferred to Appendix E.1.2. See Remark 32.

- *If $n \geq 2m$, then*

$$T_{2m}^{(n)}(x) = \sum_{i \in [q]} \mathbb{E}[h_i] a_i^{\circ 2m} = \big[\big[\mathrm{Diag}_{2m}(\mathbb{E}[h]); \overbrace{A, A, \ldots, A}^{2m \text{ times}}\big]\big]. \tag{5.20}$$

The tensor representation in (5.19) is a specific type of tensor decomposition which is a special case of the Tucker representation (since S_r is not fully dense), but more general than the CP representation. The tensor representation in (5.20) has a CP form.

5.6.2.3 Comparison with Single Topic Model and Bag-of-words Admixture Model

We now provide the tensor form for the special cases single topic model and bag-of-words admixture model. In order to have a fair comparison, the number of observed variables is fixed to $2m$ and the persistence level is varied.

CP representation of the single topic model: The $(2m)$-th order moment of the words for the single topic model (infinite-persistent topic model) is provided in equation (5.20) as

$$T_{2m}^{(\infty)}(x) = \sum_{i \in [q]} \mathbb{E}[h_i] a_i^{\circ 2m} = \big[\big[\mathrm{Diag}_{2m}(\mathbb{E}[h]); \overbrace{A, A, \ldots, A}^{2m \text{ times}}\big]\big]. \tag{5.21}$$

This representation is the symmetric CP representation of $T_{2m}^{(\infty)}(x)$. In Appendix E.3, we provide a more detailed comparison between our approach and some of the previous identifiability results for the (overcomplete) CP decomposition. In particular, we show that our uniqueness result for CP decomposition is the sparse analogue of uniqueness result in Lathauwer [116] where the factors of CP tensor decomposition (the columns of matrix A) satisfy specific sparsity constraints. See Appendix E.3 for the details.

Tucker representation of the bag-of-words admixture model: From Lemma 3, the tensor form of the $(2m)$-th order moment of observed variables $x_l, l \in [2m]$, for the bag-of-words admixture model

(1-persistent topic model) is given by

$$T_{2m}^{(1)}(x) = \sum_{i_1=1}^{q} \sum_{i_2=1}^{q} \cdots \sum_{i_{2m}=1}^{q} \mathbb{E}[h_{i_1} h_{i_2} \cdots h_{i_{2m}}] a_{i_1} \circ a_{i_2} \circ \cdots \circ a_{i_{2m}}$$
$$= \Big[\Big[\mathbb{E}\big[h^{\circ(2m)}\big]; \overbrace{A, A, \ldots, A}^{2m \text{ times}}\Big]\Big]. \tag{5.22}$$

This representation is the Tucker representation (decomposition) of $T_{2m}^{(1)}(x)$ where the core tensor $S = \mathbb{E}\big[h^{\circ(2m)}\big]$ is the tensor form of the $(2m)$-th order hidden moment $M_{2m}(h)$, defined in equation (5.3), and the inverse factors correspond to the population structure A.

Comparing the tensor forms for the n-persistent topic model (5.19), single topic model (5.21), and bag of words admixture model (5.22), we find that all of them involve Tucker decompositions, where the inverse factors correspond to the topic-word matrix A, and the only difference is in the sparsity level of the core tensor S. For the bag of words model, with $n = 1$, the core tensor is fully dense in general, while for the single topic model, with $n \to \infty$, the core tensor is diagonal which reduces to the CP decomposition. For a general topic model with persistence level n, the core tensor is in between these two extremes and has structured sparsity. This sparsity property of the core tensor is crucial towards establishing identifiability in the overcomplete regime. The bag-of-words model is not identifiable in the overcomplete regime since the core tensor is fully dense in this case, while an overcomplete n-persistent topic model can be identified under certain constraints provided in Section 5.5, since the core tensor has structured sparsity and symmetry.

5.7 Proof Techniques and Auxiliary Results

The main identifiability results are given in Theorems 5.1 and 5.2 for deterministic and random cases of topic-word graph structures. In this section, we provide a proof sketch of these results, and then, we propose auxiliary results on the existence of perfect n-gram matching for random bipartite graphs and a lower bound on the Kruskal rank of random matrices.

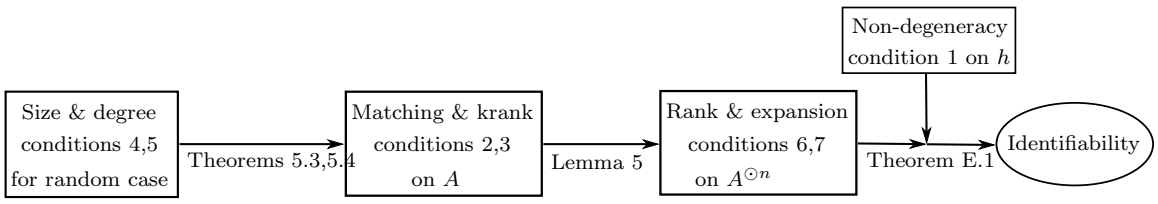


Figure 5.5: Proof outline: flow of conidtions and results

5.7.1 Proof Sketch

Summary of relationships among different conditions: To summarize, there exists a hierarchy among the proposed conditions as follows. See Figure 5.5. First, in the random analysis, the size and the degree conditions 4 and 5 are sufficient for satisfying the perfect n-gram matching and the krank conditions 2 and 3, shown by Theorems 5.3 and 5.4. Then, these conditions 2 and 3 ensure that the rank and the expansion conditions 6 and 7 hold, shown by Lemma 5. And finally, these conditions 6 and 7 together with non-degeneracy condition 1 conclude the primary identifiability result in Theorem E.1. Note that the genericity of A is also required for these results to hold.

Primary deterministic analysis in Theorem E.1: The deterministic analysis is primarily based on conditions on the n-gram matrix $A^{\odot n}$; but since these conditions are opaque (mainly expansion condition on $A^{\odot n}$, provided in condition 7), this analysis is related to conditions on the matrix A itself (see Lemma 5). See Theorem E.1 in Appendix E.1.1 for the identifiability result based on $A^{\odot n}$. We briefly discuss it below for the case when $2n$ words are available under the n-persistent topic model. From equation (5.8), the $(2n)$-th order moment of the observed variables under the n-persistent topic model can be written as

$$M_{2n}^{(n)}(x) = \Big(A^{\odot n}\Big)\mathbb{E}\big[hh^\top\big]\Big(A^{\odot n}\Big)^\top. \tag{5.23}$$

The question is whether we can recover A, given the $M_{2n}^{(n)}(x)$. Obviously, the matrix A is not identifiable without any further conditions. First, non-degeneracy and rank conditions (conditions 1 and 6) are required. Assuming these two conditions, we have from (5.23) that

$$\mathrm{Col}\Big(M_{2n}^{(n)}(x)\Big) = \mathrm{Col}\Big(A^{\odot n}\Big).$$

Therefore, the problem of recovering A from $M_{2n}^{(n)}(x)$ reduces to finding $A^{\odot n}$ in $\mathrm{Col}\big(A^{\odot n}\big)$. Then, we show that under the following expansion condition on $A^{\odot n}$ and the genericity property, matrix A is identifiable from $\mathrm{Col}\big(A^{\odot n}\big)$. The expansion condition (refer to condition 7 for a more detailed statement), imposes the following property on the bipartite graph [14] $G\big(V_h, V_o^{(n)}; A^{\odot n}\big)$,

$$\Big|N_{A^{\odot n}_{\text{Rest.}}}(S)\Big| \geq |S| + d_{\max}\Big(A^{\odot n}\Big), \quad \forall S \subseteq V_h,\ |S| > \mathrm{krank}(A), \tag{5.24}$$

where $d_{\max}\big(A^{\odot n}\big)$ is the maximum node degree in set V_h, and the restricted version of n-gram matrix, denoted by $A^{\odot n}_{\text{Rest.}}$, is obtained by removing its redundant (identical) rows (see Definition E.1). The identifiability claim is proved by showing that the columns of $A^{\odot n}$ are the sparsest and rank-1 vectors (in the tensor form) in $\mathrm{Col}\big(A^{\odot n}\big)$ under the expansion condition in (5.24) and genericity conditions. Note that since we only require expansion on sets larger than Kruskal rank, the expansion condition (5.24) is a more relaxed condition compared to expansion condition proposed in [9, 148] for identifiability in the undercomplete regime. For a more detailed comparison, refer to Remark 31 in Appendix E.1.1.

Deterministic analysis in Theorem 5.1: Expansion and rank conditions in Theorem E.1 are imposed on the n-gram matrix $A^{\odot n}$. According to the generalized matching notions, defined in Section 5.5.1, sufficient combinatorial conditions on matrix A (conditions 2 and 3) are introduced which ensure that the expansion and rank conditions on $A^{\odot n}$ are satisfied.

Recall Lemma 1 which says that existence of perfect n-gram matching in $G(Y, X; A)$ (condition 2) implies that $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching. Then, it is straightforward to argue that the expansion and rank conditions on $A^{\odot n}$ are satisfied, which is shown in Lemma 5 in Appendix E.1.3. This leads to the generic identifiability result stated in Theorem 5.1.

5.7.2 Analysis of Random Structures

The identifiability result for a random structured matrix A is provided in Theorem 5.2. Sufficient size and degree conditions 4 and 5 on the random matrix A are proposed such that the deterministic

[14] $V_o^{(n)}$ denotes all ordered n-tuples generated from set $V_o := \{1, \ldots, p\}$ which indexes the rows of $A^{\odot n}$.

combinatorial conditions 2 and 3 on A are satisfied. The details of these auxiliary results are provided in the following two subsequent sections.[15] In Section 5.7.2.1, it is shown in Theorem 5.3 that a random bipartite graph satisfying reasonable size and degree constraints, has a perfect n-gram matching (condition 2), **whp**. Then, a lower bound on the Kruskal rank of a random matrix A under size and degree constraints is provided in Theorem 5.4 in Section 5.7.2.2, which implies the krank condition 3. Intuitions on why such size and degree conditions are required, are mentioned in Section 5.5.2 where these conditions are proposed.

5.7.2.1 Existence of Perfect n-gram Matching for Random Bipartite Graphs

We show in the following theorem that a random bipartite graph satisfying reasonable size and degree constraints, proposed earlier in conditions 4 and 5, has a perfect n-gram matching **whp**.

Theorem 5.3 (Existence of perfect n-gram matching for random bipartite graphs)**.** *Consider a random bipartite graph $G(Y, X; E)$ with $|Y| = q$ nodes on the left side and $|X| = p$ nodes on the right side, and each node $i \in Y$ is randomly connected to d_i different nodes in X. Let $d_{\min} := \min_{i \in Y} d_i$. Assume that it satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ (condition 4) for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq \max\{1 + \beta \log p, \alpha \log p\}$ for some constants $\beta > \frac{n-1}{\log 1/c}, \alpha > \max\left\{2n^2\left(\beta \log \frac{1}{c} + 1\right), 2\beta n\right\}$ (lower bound in condition 5). Then, there exists a perfect (Y-saturating) n-gram matching in the random bipartite graph $G(Y, X; E)$, with probability at least $1 - \gamma_1 p^{-\beta'}$ for constants $\beta' > 0$ and $\gamma_1 > 0$, specified in (5.5) and (5.6).*

See Appendix E.2.1 for the proof.

Note that the sufficient size bound $q = O(p^n)$ in the above theorem is also necessary (see Remark 18), and is therefore tight.

Remark 25 (Insufficiency of the union bound argument). It is easier to exploit the union bound arguments to propose random bipartite graphs which have a perfect n-gram matching **whp**. It is proved in Appendix E.2.1 that if $d \geq n$ and the size constraint $|Y| = O(|X|^{\frac{n}{2}-\delta})$ for some $\delta > 0$

[15] Since these auxiliary results can also have independent interests as combinatorial results, we put them as theorems in the main part of the chapter.

is satisfied, then **whp**, the random bipartite graph has a perfect n-gram matching. Comparing this result with ours in Theorem 5.3, our approach has a better size scaling while the union bound approach has a better degree scaling. The size scaling limitation in the union bound argument makes it unattractive. In order to identify the population structure A in the overcomplete regime where $|Y| = O(|X|^n)$, we need access to at least $(4n)$-th order moment under the union bound argument, while only the $(2n)$-th order moment is required under our argument.

5.7.2.2 Lower Bound on the Kruskal Rank of Random Matrices

In the following theorem, a lower bound on the Kruskal rank of a random matrix A under dimension and degree constraints is provided.

Theorem 5.4 (Lower bound on the Kruskal rank of random matrices)**.** *Consider a random matrix $A \in \mathbb{R}^{p\times q}$, where for any $i \in [q]$, there are d_i number of random non-zero entries in column i. Let $d_{\min} := \min_{i\in[q]} d_i$. Assume that it satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ (condition 4) for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq 1 + \beta \log p$ for some constant $\beta > \frac{n-1}{\log 1/c}$ (lower bound in condition 5) and in addition A is generic. Then,* $\operatorname{krank}(A) \geq cp$*, with probability at least $1 - \gamma_2 p^{-\beta'}$ for constants $\beta' > 0$ and $\gamma_2 > 0$, specified in* (5.5) *and* (5.7)*.*

See Appendix E.2.1 for the proof.

Bibliography

[1] Evrim Acar, Seyit A Çamtepe, Mukkai S Krishnamoorthy, and Bülent Yener. Modeling and multiway analysis of chatroom tensors. In *Intelligence and Security Informatics*, pages 256–268. Springer, 2005.

[2] Radosław Adamczak, Rafał Latała, Alexander E Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Chevet type inequality and norms of submatrices. *arXiv preprint arXiv:1107.4066*, 2011.

[3] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon. Learning Sparsely Used Overcomplete Dictionaries via Alternating Minimization. *Available on arXiv:1310.7991*, Oct. 2013.

[4] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon. Learning Sparsely Used Overcomplete Dictionaries. In *COLT*, June 2014.

[5] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data generating distribution. *arXiv preprint arXiv:1211.4246*, 2012.

[6] Elizabeth S. Allman, John A. Rhodes, and Amelia Taylor. A semialgebraic description of the general markov model on phylogenetic trees. *Arxiv preprint arXiv:1212.1200*, Dec. 2012.

[7] E.S. Allman, C. Matias, and J.A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132, 2009.

[8] A. Anandkumar, D. P. Foster, D. Hsu, S. M. Kakade, and Y. K. Liu. A Spectral Algorithm for Latent Dirichlet Allocation. In *Proc. of Neural Information Processing (NIPS)*, Dec. 2012.

[9] A. Anandkumar, D. Hsu, A. Javanmard, and S. M. Kakade. Learning Linear Bayesian Networks with Latent Variables. *ArXiv e-prints*, September 2012.

[10] A. Anandkumar, D. Hsu, and S. M. Kakade. A Method of Moments for Mixture Models and Hidden Markov Models. In *Proc. of Conf. on Learning Theory*, June 2012.

[11] A. Anandkumar, D. P. Foster, D. Hsu, S. M. Kakade, and Y. K. Liu. Two SVDs Suffice: Spectral Decompositions for Probabilistic Topic Modeling and Latent Dirichlet Allocation. *to appear in the special issue of Algorithmica on New Theoretical Challenges in Machine Learning*, July 2013.

[12] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A Tensor Spectral Approach to Learning Mixed Membership Community Models. In *Conference on Learning Theory (COLT)*, June 2013.

[13] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A Tensor Spectral Approach to Learning Mixed Membership Community Models. In *Conference on Learning Theory (COLT)*, June 2013.

[14] A. Anandkumar, D. Hsu, M. Janzamin, and S. M. Kakade. When are Overcomplete Topic Models Identifiable? Uniqueness of Tensor Tucker Decompositions with Structured Sparsity. In *Neural Information Processing (NIPS)*, Dec. 2013.

[15] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor Methods for Learning Latent Variable Models. *J. of Machine Learning Research*, 15:2773–2832, 2014.

[16] A. Anandkumar, R. Ge, and M. Janzamin. Learning Overcomplete Latent Variable Models through Tensor Methods. In *Proceedings of the Conference on Learning Theory (COLT)*, Paris, France, July 2015.

[17] A. Anandkumar, R. Ge, and M. Janzamin. Learning Overcomplete Latent Variable Models through Tensor Methods. In *COLT*, Paris, France, July 2015.

[18] Anima Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014.

[19] Anima Anandkumar, Rong Ge, and Majid Janzamin. Guaranteed Non-Orthogonal Tensor Decomposition via Alternating Rank-1 Updates. *arXiv preprint arXiv:1402.5180*, Feb. 2014.

[20] Anima Anandkumar, Rong Ge, and Majid Janzamin. Sample Complexity Analysis for Learning Overcomplete Latent Variable Models through Tensor Methods. *arXiv preprint arXiv:1408.0553*, Aug. 2014.

[21] Anima Anandkumar, Rong Ge, and Majid Janzamin. Sample Complexity Analysis for Learning Overcomplete Latent Variable Models through Tensor Methods. *arXiv preprint arXiv:1408.0553*, Aug. 2014.

[22] Anima Anandkumar, Rong Ge, and Majid Janzamin. Guaranteed Non-Orthogonal Tensor Decomposition via Alternating Rank-1 Updates. *arXiv preprint arXiv:1402.5180*, Feb. 2014.

[23] J. Anderson, M. Belkin, N. Goyal, L. Rademacher, and J. Voss. The More, the Merrier: the Blessing of Dimensionality for Learning Large Gaussian Mixtures. *arXiv preprint arXiv:1311.2891*, Nov. 2013.

[24] Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1908–1916, 2014.

[25] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[26] Carl J Appellof and ER Davidson. Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents. *Analytical Chemistry*, 53(13):2053–2056, 1981.

[27] S. Arora, R. Ge, and A. Moitra. New Algorithms for Learning Incoherent and Overcomplete Dictionaries. *ArXiv e-prints*, August 2013.

[28] Saneev Arora, Rong Ge, and Ankur Moitra. Learning topic models—going beyond svd. In *Symposium on Theory of Computing*, 2012.

[29] Sanjeev Arora and Ravi Kannan. Learning mixtures of separated nonspherical gaussians. *The Annals of Applied Probability*, 15(1A):69–92, 2005.

[30] Sanjeev Arora, Rong Ge, Yoni Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. *ArXiv 1212.4777*, 2012.

[31] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. *arXiv preprint arXiv:1310.6343*, 2013.

[32] Antonio Auffinger, Gerard Ben Arous, et al. Complexity of random smooth functions on the high-dimensional sphere. *The Annals of Probability*, 41(6):4214–4247, 2013.

[33] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[34] Boaz Barak, Jonathan Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. *arXiv preprint arXiv:1407.1543*, 2014.

[35] Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993.

[36] Andrew R Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14:115–133, 1994.

[37] Peter Bartlett and Shai Ben-David. Hardness results for neural network approximation problems. In *Computational Learning Theory*, pages 50–62. Springer, 1999.

[38] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *Information Theory, IEEE Transactions on*, 44(2):525–536, 1998.

[39] Mohsen Bayati and Andrea Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *arXiv preprint arXiv:1001.3448*, Jan. 2010.

[40] Y. Bengio, A. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *arXiv preprint arXiv:1206.5538*, 2012.

[41] A. Bhaskara, M. Charikar, A. Moitra, and A. Vijayaraghavan. Smoothed analysis of tensor decompositions. *arXiv preprint arXiv:1311.3651*, 2013.

[42] A. Bhaskara, M. Charikar, and A. Vijayaraghavan. Uniqueness of Tensor Decompositions with Applications to Polynomial Identifiability. *ArXiv 1304.8087*, April 2013.

[43] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[44] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. In *Machine learning: From theory to applications*, pages 9–28. Springer, 1993.

[45] Cristiano Bocci, Luca Chiantini, and Giorgio Ottaviani. Refined methods for the identifiability of tensors. *arXiv preprint arXiv:1303.6915*, 2013.

[46] Martin L Brady, Raghu Raghavan, and Joseph Slawny. Back propagation fails to separate where perceptrons succeed. *Circuits and Systems, IEEE Transactions on*, 36(5):665–674, 1989.

[47] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006.

[48] J. F. Cardoso and Pierre Comon. Independent component analysis, a survey of some algebraic methods. In *IEEE International Symposium on Circuits and Systems*, pages 93–96, 1996.

[49] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3): 283–319, 1970.

[50] Dustin Cartwright and Bernd Sturmfels. The number of eigenvalues of a tensor. *Linear Algebra and its Applications*, 438(2):942–952, January 2013.

[51] V. Chandrasekaran, P. Parrilo, A. Willsky, et al. Latent variable graphical model selection via convex optimization. In *2010 48th Annual Allerton Conference*, pages 1610–1613. IEEE, 2010.

[52] J.T. Chang. Full reconstruction of markov models on evolutionary trees: identifiability and consistency. *Mathematical Biosciences*, 137(1):51–73, 1996.

[53] Luca Chiantini and Giorgio Ottaviani. On generic identifiability of 3-tensors of small rank. *SIAM Journal on Matrix Analysis and Applications*, 33(3):1018–1037, 2012.

[54] Luca Chiantini, Massimiliano Mella, and Giorgio Ottaviani. One example of general unidentifiable tensors. *arXiv preprint arXiv:1303.6914*, 2013.

[55] A. Choromanska, M. Henaff, M. Mathieu, G. Arous, and Y. LeCun. The loss surface of multilayer networks. In *AISTATS*, 2015.

[56] V. Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.

[57] A. Coates, H. Lee, and A. Y. Ng. An analysis of single-layer networks in unsupervised feature learning. *Journal of Machine Learning Research - Proceedings Track*, 15:215–223, 2011.

[58] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[59] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3): 287–314, 1994.

[60] P. Comon. Tensor decompositions. *Mathematics in Signal Processing V*, pages 1–24, 2002.

[61] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press. Elsevier, 2010.

[62] P. Comon, X. Luciani, and A. De Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7-8):393–405, 2009.

[63] G. Cybenko. approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[64] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[65] Sanjoy Dasgupta. Learning mixutres of gaussians. In *FOCS*, 1999.

[66] Sanjoy Dasgupta, Daniel Hsu, and Nakul Verma. A concentration theorem for projections. In *Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006.

[67] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *NIPS*, pages 2933–2941, 2014.

[68] L. De Lathauwer, J. Castaing, and J.-F Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Tran. on Signal Processing*, 55:2965–2973, June 2007.

[69] L. De Lathauwer, J. Castaing, and J.-F. Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *Signal Processing, IEEE Transactions on*, 55(6): 2965–2973, 2007.

[70] Lieven De Lathauwer and Joséphine Castaing. Blind identification of underdetermined mixtures by simultaneous matrix diagonalization. *Signal Processing, IEEE Transactions on*, 56 (3):1096–1105, 2008.

[71] Li Deng and Dong Yu. *Deep Learning for Signal and Information Processing*. NOW Publishers, 2013.

[72] Ignat Domanov and Lieven De Lathauwer. On the uniqueness of the canonical polyadic decomposition of third-order tensors—part i: Basic results and uniqueness of one factor matrix. *SIAM Journal on Matrix Analysis and Applications*, 34(3):855–875, 2013.

[73] Ignat Domanov and Lieven De Lathauwer. On the uniqueness of the canonical polyadic decomposition of third-order tensors—part ii: Uniqueness of the overall decomposition. *SIAM Journal on Matrix Analysis and Applications*, 34(3):876–903, 2013.

[74] D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[75] P Frasconi, M Gori, and A Tesi. Successes and failures of backpropagation: A theoretical investigation. *Progress in Neural Networks: Architecture*, 5:205, 1997.

[76] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[77] Rong Ge and Tengyu Ma. Decomposing overcomplete 3rd order tensors using sum-of-squares algorithms. *arXiv preprint arXiv:1504.05287*, April 2015.

[78] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 1990.

[79] Marco Gori and Alberto Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.

[80] N. Goyal, S. Vempala, and Y. Xiao. Fourier pca. *arXiv preprint arXiv:1306.5825*, 2013.

[81] Olivier Guédon and Mark Rudelson. Lp-moments of random vectors via majorizing measures. *Advances in Mathematics*, 208(2):798–823, 2007.

[82] Benjamin D. Haeffele and René Vidal. Global optimality in tensor factorization, deep learning, and beyond. *CoRR*, abs/1506.07540, 2015.

[83] Philip Hall. On representatives of subsets. *J. London Math. Soc.*, 10(1):26–30, 1935.

[84] Moritz Hardt. On the provable convergence of alternating minimization for matrix completion. *arXiv preprint arXiv:1312.0925*, 2013.

[85] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

[86] Richard A Harshman. Foundations of the parafac procedure: models and conditions for an" explanatory" multimodal factor analysis. 1970.

[87] Richard A Harshman and Margaret E Lundy. Parafac: Parallel factor analysis. *Computational Statistics & Data Analysis*, 18(1):39–72, 1994.

[88] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP hard. *arXiv preprint arXiv:0911.1393*, 2009.

[89] Christopher J Hillar and Friedrich T Sommer. Ramsey theory reveals the conditions when sparse coding on subsampled data is unique. *arXiv preprint arXiv:1106.3616*, 2011.

[90] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[91] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[92] K. Hornik, M. Stinchcombe, and H. White. multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[93] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[94] D. Hsu and S. M. Kakade. Learning Mixtures of Spherical Gaussians: Moment Methods and Spectral Decompositions. *arXiv preprint arXiv:1206.5766*, 2012.

[95] Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 11–20. ACM, 2013.

[96] Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.

[97] F. Huang, U. N. Niranjan, M. Hakeem, and A. Anandkumar. Fast Detection of Overlapping Communities via Online Tensor Methods. *ArXiv 1309.0787*, Sept. 2013.

[98] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[99] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. In *Journal of Machine Learning Research*, pages 695–709, 2005.

[100] Piotr Indyk and Ilya Razenshteyn. On model-based RIP-1 matrices. *CoRR*, abs/1304.3604, 2013. URL `http://arxiv.org/abs/1304.3604`.

[101] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 665–674. ACM, 2013.

[102] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Score Function Features for Discriminative Learning: Matrix and Tensor Frameworks. *arXiv preprint arXiv:1412.2863*, Dec. 2014.

[103] Tao Jiang and Nicholas D Sidiropoulos. Kruskal's permutation lemma and the identification of candecomp/parafac and bilinear models with constant modulus constraints. *Signal Processing, IEEE Transactions on*, 52(9):2625–2636, 2004.

[104] A. T. Kalai, A. Moitra, and G. Valiant. Efficiently learning mixtures of two gaussians. In *STOC*, 2010.

[105] Michael J Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[106] M. Amin Khajehnejad, Alexandros G. Dimakis, Weiyu Xu, and Babak Hassibi. Sparse recovery of nonnegative signals with minimal expansion. *IEEE Transactions on Signal Processing*, 59(1):196–208, 2011.

[107] T. Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2001.

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

[109] Pravesh Kothari and Raghu Meka. Almost optimal pseudorandom generators for spherical caps. *arXiv preprint arXiv:1411.6299*, 2014.

[110] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15:349–396, February 2003.

[111] J.B. Kruskal. More factors than subjects, tests and treatments: an indeterminacy theorem for canonical decomposition and individual differences scaling. *Psychometrika*, 41(3):281–293, 1976.

[112] J.B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18 (2):95–138, 1977.

[113] Christian Kuhlmann. Hardness results for general two-layer neural networks. In *COLT*, pages 275–285, 2000.

[114] Joseph M Landsberg. *Tensors: Geometry and applications*, volume 128. American Mathematical Soc., 2012.

[115] R. Latala. Estimates of moments and tails of Gaussian chaoses. *Ann. Prob.*, 34(6):2315–2331, 2006.

[116] Lieven De Lathauwer. A Link between the Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization. *SIAM J. Matrix Analysis and Applications*, 28(3): 642–666, 2006.

[117] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng. ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning. In *NIPS*, pages 1017–1025, 2011.

[118] Quoc V. Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Y. Ng. ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning. In *NIPS*, pages 1017–1025, 2011.

[119] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000.

[120] R. Livni, S. Shalev-Shwartz, and O. Shamir. On the computational efficiency of training neural networks. In *NIPS*, pages 855–863, 2014.

[121] Robert J Marks II and Payman Arabshahi. Fourier analysis and filtering of a single hidden layer perceptron. In *International Conference on Artificial Neural Networks (IEEE/ENNS), Sorrento, Italy*, 1994.

[122] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. *ArxXiv preprint*, abs/1209.0738, 2012.

[123] B. McWilliams, D. Balduzzi, and J. Buhmann. Correlated random features for fast semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 440–448, 2013.

[124] Nishant A. Mehta and Alexander G. Gray. Sparsity-based generalization bounds for predictive sparse coding. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, Atlanta, USA, June 2013.

[125] Jorma K. Merikoski and Ravinder Kumar. Inequalities for spreads of matrix sums and products. *Applied Mathematics E-Notes*, 4:150–159, 2004.

[126] J Mocks. Topographic components model for event-related potentials and some biophysical considerations. *IEEE transactions on biomedical engineering*, 6(35):482–484, 1988.

[127] A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of gaussians. In *FOCS*, 2010.

[128] Nelson Morgan and Hervé Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In *NIPS*, pages 630–637, 1989.

[129] E. Mossel and S. Roch. Learning nonsingular phylogenies and hidden markov models. *The Annals of Applied Probability*, 16(2):583–614, 2006.

[130] Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. Phase retrieval using alternating minimization. *arXiv preprint arXiv:1306.0160*, 2013.

[131] N. H. Nguyen, P. Drineas, and T. D. Tran. Tensor sparsification via a bound on the spectral norm of random tensors. *arXiv preprint arXiv:1005.4732*, May 2010.

[132] XuanLong Nguyen. Posterior contraction of the population polytope in finite admixture models. *arXiv preprint arXiv:1206.0068*, 2012.

[133] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 186:343–414, 1895.

[134] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

[135] Yuval Rabani, Leonard Schulman, and Chaitanya Swamy. Learning mixtures of arbitrary distributions over large discrete domains. *arXiv preprint arXiv:1212.1527*, 2012.

[136] B. Rao and K. Kreutz-Delgado. An affine scaling methodology for best basis selection. *IEEE Tran. Signal Processing*, 47:187–200, January 1999.

[137] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 1996.

[138] M. Rudelson and R. Vershynin. The smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

[139] Mark Rudelson and Roman Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

[140] Hanie Sedghi and Anima Anandkumar. Provable methods for training neural networks with sparse connectivity. *NIPS workshop on Deep Learning and Representation Learning*, Dec. 2014.

[141] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[142] Amnon Shashua and Anat Levin. Linear image coding for regression and classification using the tensor-rank principle. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–42. IEEE, 2001.

[143] Nicholas D. Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of N-way arrays. *Journal of Chemometrics*, 14(3):229–239, 2000.

[144] Nicholas D Sidiropoulos, Rasmus Bro, and Georgios B Giannakis. Parallel factor analysis in sensor array processing. *Signal Processing, IEEE Transactions on*, 48(8):2377–2388, 2000.

[145] Jiří Šíma. Training a single sigmoidal neuron is hard. *Neural Computation*, 14(11):2709–2728, 2002.

[146] Matthew Skala. Hypergeometric tail inequalities: ending the insanity. `http://ansuz.sooke.bc.ca/professional/hypergeometric.pdf`.

[147] L. Song, A. Anandkumar, B. Dai, and B. Xie. Nonparametric estimation of multi-view latent variable models. *Available on arXiv:1311.3287*, Nov. 2013.

[148] Daniel A. Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. *ArxXiv preprint*, abs/1206.5882, 2012.

[149] Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Proc. of Conf. on Learning Theory*, 2012.

[150] Bharath Sriperumbudur, Kenji Fukumizu, Revant Kumar, Arthur Gretton, and Aapo Hyvärinen. Density estimation in infinite dimensional exponential families. *arXiv preprint arXiv:1312.3516*, 2013.

[151] Alwin Stegeman, Jos M.F. Ten Berge, and Lieven De Lathauwer. Sufficient conditions for uniqueness in candecomp/parafac and indscal with random component matrices. *Psychometrika*, 71(2):219–229, June 2006.

[152] Kevin Swersky, David Buchman, Nando D Freitas, Benjamin M Marlin, et al. On autoencoders and score matching for energy based models. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1201–1208, 2011.

[153] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[154] André Uschmajew. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 33(2):639–652, 2012.

[155] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–93. IEEE, 2003.

[156] S. Vempala and G. Wang. A spectral algorithm for learning mixtures of distributions. In *FOCS*, 2002.

[157] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[158] Yining Wang, Hsiao-Yu Tung, Alexander Smola, and Anima Anandkumar. Fast and guaranteed tensor decomposition via sketching. In *Proc. of NIPS*, 2015.

[159] Eugene P Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *The Annals of Mathematics*, 62(3):548–564, 1955.

[160] T. Zhang and G. Golub. Rank-one approximation to high order tensors. *SIAM Journal on Matrix Analysis and Applications*, 23:534–550, 2001.

[161] Yuchen Zhang, Jason D. Lee, and Michael I. Jordan. ℓ_1-regularized neural networks are improperly learnable in polynomial time. *CoRR*, abs/1510.03528, 2015.

[162] J. Y. Zou, D. Hsu, D. C. Parkes, and R. P. Adams. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems*, pages 2238–2246, 2013.

Appendix A

Proofs for Overcomplete CP Tensor Decomposition: Incoherent Components

A.1 More Related Works

Tensor decomposition for learning undercomplete models: Several latent variable models can be learned through tensor decomposition including independent component analysis [69], topic models, Gaussian mixtures, hidden Markov models [15] and network community models [12]. In the undercomplete setting, Anandkumar et al. [15] analyze robust tensor power iteration for learning LVMs, and Song et al. [147] extend analysis to the nonparametric setting. These works require the tensor factors to have full column rank, which rules out overcomplete models. Moreover, they require whitening the input data, and hence the sample complexity depends on the condition number of the factor matrices. For instance, when $k = d$, for random factor matrices, the previous tensor approaches in Song et al. [147], Anandkumar et al. [11] have a sample complexity of $\tilde{\Omega}(k^{6.5})$, while our result provides improved sample complexity $\tilde{\Omega}(k^2)$ assuming incoherent components.

Learning overcomplete models: In general, learning overcomplete models is challenging, and they may not even be identifiable. The FOOBI procedure by De Lathauwer et al. [69] shows that a polynomial-time procedure can recover the components of ICA model (with *generic* factors) when $k = O(d^2)$, where the moment is fourth order. However, the procedure does not work for third-order overcomplete tensors. For the fifth order tensor, Goyal et al. [80], Bhaskara et al. [41] perform simultaneous diagonalization on the matricized versions of random slices of the tensor and provide careful perturbation analysis. But, this procedure cannot handle the same level of overcompleteness as FOOBI, since an additional dimension is required for obtaining two (or more) fourth order tensor slices. In addition, Goyal et al. [80] provide stronger results for ICA, where the tensor slices can be obtained in the Fourier domain. Given 4th order tensor, they need poly(k^4) number of unlabeled samples for learning ICA (where the poly factor is not explicitly characterized), while we only need $\tilde{\Omega}(k^{2.5})$ (when $k = \Theta(d^2)/\operatorname{polylog}(d)$). Anderson et al. [23] convert the problem of learning Gaussian mixtures to an ICA problem and exploit the Fourier PCA method in Goyal et al. [80]. More precisely, for a Gaussian mixtures model with known identical covariance matrices, when the number of components $k = \operatorname{poly}(d)$, the model can be learned in polynomial time (as long as a certain non-degeneracy condition is satisfied).

Arora et al. [27], Agarwal et al. [3], Barak et al. [34] provide guarantees for the sparse coding model (also known as dictionary learning problem). Arora et al. [27], Agarwal et al. [3] provide clustering based approaches for approximately learning incoherent dictionaries and then refining them through alternating minimization to obtain exact recovery of both the dictionary and the coefficients. They can handle sparsity level up to $O(\sqrt{d})$ (per sample) and the size of the dictionary k can be arbitrary. Barak et al. [34] consider tensor decomposition and dictionary learning using sum-of-squares (SOS) method. In contrast to simple iterative updates considered here, SOS involves solving semi-definite programs. They provide guaranteed recovery by a polynomial time complexity $k^{O(1/\delta)}$ for some $0 < \delta < 1$, when the size of the dictionary $k = \Theta(d)$, and the sparsity level is $k^{1-\delta}$. They also provide guarantees for higher sparsity levels up to (a small enough) constant fraction of k, but the computational complexity of the algorithm becomes quasi-polynomial: $k^{O(\log k)}$. They can also handle higher level of overcompleteness at the expense of reduced sparsity level. They do not require any incoherence conditions on the factor matrices and they can handle the signal to

noise ratio being a constant. Thus, their work has strong guarantees, but at the expense of running a complicated algorithm. In contrast, we consider a simple alternating rank-1 updates algorithm, but require more stringent conditions on the model.

There are other recent works which can learn overcomplete models, but under different settings than the one considered in this work. Anandkumar et al. [14] learn overcomplete sparse topic models, and provide guarantees for *Tucker* tensor decomposition under sparsity constraints. Specifically, the model is identifiable using $(2n)^{\text{th}}$ order moments when the latent dimension $k = O(d^n)$ and the sparsity level of the factor matrix is $O(d^{1/n})$, where d is the observed dimension. The Tucker decomposition is more general than the CP decomposition considered here, and the techniques in [14] differ significantly from the ones considered here, since they incorporate sparsity, while we incorporate incoherence here.

Concentration Bounds: We obtain tight concentration bounds for empirical tensors in this work. In contrast, applying matrix concentration bounds, e.g. [153], leads to strictly worse bounds since they require matricizations of the tensor. Latala [115] provides an upper bound on the moments of the Gaussian chaos, but they are limited to independent Gaussian distributions (and can be extended to other cases such as Rademacher distribution). The principle of entropy-concentration trade-off [138], employed in this work, have been used in other contexts. For instance, Nguyen et al. [131] provide a spectral norm bound for random tensors. They first apply a symmetrization argument which reduces the problem to bounding the spectral norm of a random Gaussian tensor and then employ entropy-concentration trade-off to bound its spectral norm. They also exploit the bounds on the Lipschitz functions of Gaussian random variables. While Nguyen et al. [131] employ a rough classification of vectors (to be covered) into dense and sparse vectors, we require a finer classification of vectors into different "buckets" (based on their inner products with given vectors) to obtain the tight concentration bounds in this work. Moreover, we do not impose Gaussian assumption in this work, and instead require more general conditions such as RIP or bounded 2-to-3 norms.

A.2 Deterministic Assumptions

In the main text, we assume matrices A, B, and C are randomly generated. However, we are not using all the properties of randomness. In particular, we only need the following assumptions.

(A1) **Rank-k decomposition:** The third order tensor T has a CP rank of $k \geq 1$ with decomposition

$$T = \sum_{i \in [k]} w_i (a_i \otimes b_i \otimes c_i), \quad w_i > 0, a_i, b_i, c_i \in \mathcal{S}^{d-1}, \forall\, i \in [k], \tag{A.1}$$

where $\mathcal{S}^{d-1}$ denotes the unit d-dimensional sphere, i.e. all the vectors have unit[1] 2-norm as $\|a_i\| = \|b_i\| = \|c_i\| = 1, i \in [k]$. Furthermore, define $w_{\min} := \min_{i \in [k]} w_i$ and $w_{\max} := \max_{i \in [k]} w_i$.

(A2) **Incoherence:** The components are incoherent, and let

$$\rho := \max_{i \neq j} \{|\langle a_i, a_j \rangle|, |\langle b_i, b_j \rangle|, |\langle c_i, c_j \rangle|\} \leq \frac{\alpha}{\sqrt{d}}, \tag{A.2}$$

for some $\alpha = \text{polylog}(d)$. In other words, $A^\top A = I + J_A$, $B^\top B = I + J_B$, and $C^\top C = I + J_C$, where J_A, J_B, and J_C, are incoherence matrices with zero diagonal entries. We have $\max\{\|J_A\|_\infty, \|J_B\|_\infty, \|J_C\|_\infty\} \leq \rho$ as in (A.2).

(A3) **Spectral norm conditions:** The components satisfy spectral norm bound

$$\max\{\|A\|, \|B\|, \|C\|\} \leq 1 + \alpha_0 \sqrt{\frac{k}{d}},$$

for some constant $\alpha_0 > 0$.

[1] This normalization is for convenience and the results hold for general case.

(A4) **Bounds on tensor norms:** Tensor T satisfies the bound

$$\|T\| \le w_{\max}\alpha_0,$$
$$\left\|T_{\setminus j}(a_j, b_j, I)\right\| := \left\|\sum_{i\neq j} w_i \langle a_i, a_j\rangle\langle b_i, b_j\rangle c_j\right\| \le \alpha w_{\max}\frac{\sqrt{k}}{d},$$

for some constant α_0 and $\alpha = \text{polylog}(d)$.

(A5) **Rank constraint:** The rank of the tensor is bounded by $k = o\left(d^{1.5}/\operatorname{polylog} d\right)$.

(A6) **Bounded perturbation:** Let ψ denote the spectral norm of perturbation tensor as

$$\psi := \|\Psi\|. \tag{A.3}$$

Suppose ψ is bounded as [2]

$$\psi \le \min\left\{\frac{1}{6}, \frac{\sqrt{\log k}}{\alpha_0\sqrt{d}}\right\} \cdot w_{\min},$$

where α_0 is a constant.

(A7) **Weights ratio:** The maximum ratio of weights $\gamma := \frac{w_{\max}}{w_{\min}}$ satisfies the bound

$$\gamma = O\left(\min\left\{\sqrt{d}, \frac{d^{1.5}}{k}\right\}\right).$$

(A8) **Contraction factor:** The contraction factor q in Theorem 2.4 is defined as

$$q := \frac{2w_{\max}}{w_{\min}}\left[\frac{2\alpha}{\sqrt{d}}\left(1 + \alpha_0\sqrt{\frac{k}{d}}\right)^2 + \beta'\right], \tag{A.4}$$

for some constants $\alpha_0, \beta' > 0$, and $\alpha = \text{polylog}(d)$. In particular, we need $\alpha\alpha_0\sqrt{k}/d + \beta' < w_{\max}/10w_{\min}$ which ensures $q < 1/2$. This is satisfied when $\sqrt{k}/d < w_{\max}/w_{\min}\operatorname{poly}\log d$ and $\beta' < w_{\max}/20w_{\min}$. The parameter β' is determined by the following assumption (initialization).

[2] Note that for the local convergence guarantee, only the first condition $\psi \le \frac{w_{\min}}{6}$ is required.

(A9) **Initialization:** Let

$$\epsilon_0 := \max\left\{\text{dist}\left(\widehat{a}^{(0)}, a_j\right), \text{dist}\left(\widehat{b}^{(0)}, b_j\right)\right\},$$

denote the initialization error w.r.t. to some $j \in [k]$. Suppose it is bounded as

$$\epsilon_0 \leq \min\left\{\frac{\beta'}{\alpha_0}, \sqrt{\frac{w_{\min}}{6w_{\max}}}, \frac{w_{\min}q}{4w_{\max}}, \frac{2w_{\max}}{w_{\min}q}\left(\frac{w_{\min}}{6w_{\max}} - \alpha\frac{\sqrt{k}}{d}\right)\right\},$$

for some constants $\alpha_0, \beta' > 0$, $\alpha = \text{polylog}(d)$, and $0 < q < 1/2$ which is defined in (A.4).

(A10) $2 \to p$ **norm:** For some fixed constant $p < 3$, $\max\{\|A^\top\|_{2\to p}, \|B^\top\|_{2\to p}, \|C^\top\|_{2\to p}\} \leq 1 + o(1)$.

Remark 26. Many of the assumptions are actually parameter choices. The only properties of random matrices required are (A2), (A3), (A4) and (A10),. See Appendix A.2.1 for detailed discussion.

Let us provide a brief discussion about the above assumptions. Condition (A1) requires the presence of a rank-k decomposition for tensor T. We normalize the component vectors for convenience, and this removes the scaling indeterminacy issues which can lead to problems in convergence. Additionally, we impose incoherence constraint in (A2), which allows us to provide convergence guarantee in the overcomplete setting. Assumptions (A3) and (A4) impose bounds on the spectral norm of tensor T and its decomposition components. Note that assumptions (A2)-(A4) and (A10) are satisfied w.h.p. when the columns of A, B, and C are generically drawn from unit sphere $\mathcal{S}^{d-1}$ (see Lemma A.1 and Guédon and Rudelson [81]), all others are parameter choices. Assumption (A5) limits the overcompleteness of problem which is required for providing convergence guarantees. The first bound on perturbation in (A6) as $\psi \leq \frac{w_{\min}}{6}$ is required for local convergence guarantee and the second bound $\psi \leq \frac{w_{\min}\sqrt{\log k}}{\alpha_0\sqrt{d}}$ is needed for arguing initialization provided by Procedure 2. Assumption (A7) is required to ensure contraction happens in each iteration. Assumption (A8) defines contraction ratio q in each iteration, and Assumption (A9) is the initialization condition required for local convergence guarantee.

The tensor-spectral norm and $2 \to p$ norm assumptions (A4) and (A10) may seem strong as we cannot even verify them given the matrix. However, when $k < d^{1.25-\epsilon}$ for arbitrary constant $\epsilon > 0$,

both conditions are implied by incoherence. See Lemma A.3. We only need these assumptions to go to the very overcomplete setting.

A.2.1 Random matrices satisfy the deterministic assumptions

Here, we provide arguments that random matrices satisfy conditions (A2), (A3), (A4), and (A10). It is well known that random matrices are incoherent, and have small spectral norm (bound on spectral norm dates back to Wigner [159]). See the following lemma.

Lemma A.1. *Consider random matrix $X \in \mathbb{R}^{d\times k}$ where its columns are uniformly drawn at random from unit d-dimensional sphere $\mathcal{S}^{d-1}$. Then, it satisfies the following incoherence and spectral bounds with high probability as*

$$\max_{i,j\in[k],i\neq j} |\langle X_i, X_j\rangle| \leq \frac{\alpha}{\sqrt{d}},$$
$$\|X\| \leq 1 + \alpha_0\sqrt{\frac{k}{d}},$$

for some $\alpha = O(\sqrt{\log k})$ and $\alpha_0 = O(1)$.

The spectral norm of the tensor is less well-understood. However, it can be bounded by the $2 \to 3$ norm of matrices. Using tools from Guédon and Rudelson [81], Adamczak et al. [2], we have the following result.

Lemma A.2. *Consider a random matrix $A \in \mathbb{R}^{d\times k}$ whose columns are drawn uniformly at random from unit sphere. If $k < d^{p/2}/\operatorname{polylog}(d)$, then*

$$\left\|A^\top\right\|_{2\to p} \leq 1 + o(1).$$

This directly implies Assumption (A10). In particular, since we only apply Assumption (A10) to unsupervised setting ($k \leq O(d)$) in Appendix A.5, for randomly generated tensor, Assumption (A10) holds for all $p > 2$ (notice that we only need it to hold for some $p < 3$).

We also give an alternative proof of $2 \to p$ norm which does not assume randomness and only relies on incoherence.

Lemma A.3. *Suppose columns of matrix $A \in \mathbb{R}^{d\times k}$ have unit norm and satisfy the incoherence condition (A2) and spectral norm condition (A3). If $k \le d^{1.25-\epsilon}$ for arbitrary constant $\epsilon > 0$, then for any $p > 3 - 2\epsilon$, we have*

$$\left\|A^\top\right\|_{2\to p} \le 1 + o(1).$$

Proof: Let $L = \sqrt{d}/\operatorname{poly}\log d$. By incoherence assumption we know every subset of L columns in A has singular values within $1 \pm o(1)$ (by Gershgorin Disk Theorem).

For any unit vector u, let S be the set of L indices that are largest in $A^\top u$. By the argument above we know $\|(A_S)^\top u\| \le \|A_S\|\|u\| \le 1 + o(1)$. In particular, the smallest entry in $A_S^\top u$ is at most $2/\sqrt{L}$. By construction of S this implies for all i not in S, $|A_i^\top u|$ is at most $2/\sqrt{L}$. Now we can write the ℓ_p ($p > 2$) norm of $A^\top u$ as

$$\begin{aligned}
\|A^\top u\|_p^p &= \sum_{i\in S} |A_i^\top u|^p + \sum_{i\notin S} |A_i^\top u|^p \\
&\le \sum_{i\in S} |A_i^\top u|^2 + (2/\sqrt{L})^{p-2} \sum_{i\notin S} |A_i^\top u|^2 \\
&\le 1 + o(1).
\end{aligned}$$

Here the first inequality uses that every entry outside S is small, and last inequality uses the bound argued on $\|(A_S)^\top u\|$, the spectral norm bound assumed on A_{S^c} and the fact that $p > 3 - 2\epsilon$. □

The $2 \to 3$ norm implies a bound on the tensor spectral norm by Hölder's inequality.

Fact 1 (Hölder's Inequality)**.** *When $1/p + 1/q = 1$, for two sequence of numbers $\{a_i\}, \{b_i\}$, we have*

$$\sum_i a_i b_i \le \left(\sum_i |a_i|^p\right)^{1/p} \left(\sum_i |b_i|^q\right)^{1/q}.$$

Consequently, we have the following corollary.

Corollary A.1. *For vectors f, g, h, and weights $w_i \geq 0$, we have*

$$\sum_i w_i f_i g_i h_i \leq w_{\max} \|f\|_3 \|g\|_3 \|h\|_3.$$

Proof: The proof applies Hölder's inequality twice as

$$\sum_i w_i f_i g_i h_i \leq w_{\max} \sum_i |f_i g_i h_i| \leq w_{\max} (\sum |f_i|^3)^{1/3} (\sum |g_i h_i|^{3/2})^{2/3} \leq w_{\max} \|f\|_3 \|g\|_3 \|h\|_3,$$

where in the first application, $p = 3$ and $q = 3/2$, and in the second application, $p = q = 2$ (which is the special case known as Cauchy-Schwartz). □

In the following lemma, it is shown that the first bound in Assumption (A4) holds for random matrices w.h.p.

Lemma A.4. *Let A, B, and C be random matrices in $\mathbb{R}^{d \times k}$ whose columns are drawn uniformly at random from unit sphere. If $k < d^{3/2}/\operatorname{polylog}(d)$, and*

$$T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i,$$

then

$$\|T\| \leq O(w_{\max}).$$

Proof: For any unit vectors $\widehat{a}, \widehat{b}, \widehat{c}$, we have

$$\begin{aligned} T(\widehat{a}, \widehat{b}, \widehat{c}) &= \sum_{i \in [k]} w_i (A^\top \widehat{a})_i (B^\top \widehat{b})_i (C^\top \widehat{c})_i \\ &\leq w_{\max} \|A^\top \widehat{a}\|_3 \|B^\top \widehat{b}\|_3 \|C^\top \widehat{c}\|_3 \\ &\leq w_{\max} \|A^\top\|_{2 \to 3} \|\widehat{a}\| \cdot \|B^\top\|_{2 \to 3} \|\widehat{b}\| \cdot \|C^\top\|_{2 \to 3} \|\widehat{c}\| \\ &= O(w_{\max}), \end{aligned}$$

where Corollary A.1 is exploited in the first inequality, and Lemma A.2 is used in the last inequality. □

For the case with two undercomplete and one overcomplete dimensions (see Corollary 2.2), we can prove the tensor spectral norm using basic properties of the matrices A, B, C.

Lemma A.5. *Let $A, B \in \mathbb{R}^{d_u \times k}$ be matrices with spectral norm bounded by $O(1)$, and $C \in \mathbb{R}^{d_o \times k}$ be a matrix whose columns have unit norm. Let*

$$T = \sum_{i=1}^{k} w_i a_i \otimes b_i \otimes c_i,$$

then we have

$$\|T\| \leq O(w_{\max}).$$

Proof: For any unit vectors $u, v \in \mathbb{R}^{d_u}$ and $w \in \mathbb{R}^{d_o}$, by assumptions we know $\|A^\top u\| \leq O(1), \|B^\top v\| \leq O(1)$ and $\|C^\top w\|_\infty \leq 1$. Now we have

$$\begin{aligned} T(u, v, w) &= \sum_{i=1}^{k} w_i \langle a_i, u\rangle \langle b_i, v\rangle \langle c_i, w\rangle \\ &\leq w_{\max} \sum_{i=1}^{k} |\langle a_i, u\rangle \langle b_i, v\rangle| \\ &\leq w_{\max} \|A^\top u\| \|B^\top v\| \\ &= O(w_{\max}). \end{aligned}$$

The first inequality uses triangle inequality and the fact that $|\langle c_i, w\rangle| \leq 1$. The Cauchy-Schwartz inequality is exploited in the second inequality. Therefore, the spectral norm of the tensor is bounded by $O(w_{\max})$. □

Finally, we show in the following lemma that the second bound in Assumption (A4) is satisfied for random matrices.

Lemma A.6. *Let $A, B, C \in \mathbb{R}^{d\times k}$ be independent, normalized (column) Gaussian matrices. Then for all $i \in [k]$, we have with high probability*

$$\left\| C_{\setminus i} \operatorname{diag}(w^{\setminus i})(J_A * J_B)_i^{\setminus i} \right\| = \tilde{O}\left(w_{\max} \frac{\sqrt{k}}{d} \right).$$

Proof: We have

$$C_{\setminus i} \operatorname{diag}(w^{\setminus i})(J_A * J_B)_i^{\setminus i} = \sum_{j\neq i} C_j w_j \langle A_i, A_j\rangle \langle B_i, B_j\rangle = \sum_{j\neq i} C_j \delta_j,$$

where $\delta_j := w_j \langle A_i, A_j\rangle \langle B_i, B_j\rangle$ is independent of C_j. From Lemma A.1, columns of A and B are incoherent, and therefore, for $j \neq i$, we have

$$|\delta_j| = \tilde{O}(w_{\max}/d).$$

Now since C_j's are independent, zero mean vectors, the sum $\sum_{j\neq i} \delta_j C_j$ is zero mean and its variance is bounded by $\tilde{O}(w_{\max}^2 k/d^2)$. Then, from vector Bernstein's bound we have with high probability

$$\left\| C_{\setminus i} \operatorname{diag}(w^{\setminus i})(J_A * J_B)_i^{\setminus i} \right\| = \tilde{O}\left(w_{\max} \frac{\sqrt{k}}{d} \right).$$

The proof is completed by applying union bound. □

Spectral norm of Khatri-Rao product

For the convergence guarantees of the second step of algorithm on removing residual error, we need the following additional bound on the spectral norm of Khatri-Rao product of random matrices.

(A11) **Spectral Norm Condition on Khatri-Rao Products:** The components satisfy the following spectral norm bound on the Khatri-Rao products as

$$\max\{\|A \odot B\|, \|B \odot C\|, \|A \odot C\|\} \leq 1 + \alpha_0 \frac{\sqrt{k}}{d},$$

for $\alpha_0 \leq \text{poly} \log d$.

We now prove that Assumption (A11) is satisfied with high probability, if the columns of A, B and C are uniformly i.i.d. drawn from unit d-dimensional sphere.

The key idea is to view $(A \odot B)^\top (A \odot B)$ as the sum of random matrices, and use the following Matrix Bernstein's inequality to prove concentration results.

Lemma A.7. *Let $M = \sum_{i=1}^n M_i$ be sum of independent symmetric $d \times d$ matrices with $\mathbb{E}[M_i] = 0$, assume all matrices M_i's have spectral norm at most R almost surely, let $\sigma^2 = \|\mathbb{E}[M_i^2]\|$, then for any τ*

$$\Pr[\|M\| \geq \tau] \leq 2d \exp\left(\frac{-\tau^2/2}{\sigma^2 + R\tau/3}\right).$$

A.2.1.0.1 Remark: Although the lemma requires all M_i's to have spectral norm at most R almost surely, it suffices to have spectral norm bounded by R with high probability and bounded by $R^\infty = \text{poly}(d, k)$ almost surely. This is because we can always condition on the fact that $\|M_i\| \leq R$ for all i. Such conditioning can only change the expectations by a negligible amount, and does not affect independence between M_i's.

Random unit vectors are not easy to work with, as entries in the same column are not independent. Thus, we first prove the result for matrices A and B whose entries are independent Gaussian variables.

Lemma A.8. *Suppose A, $B \in \mathbb{R}^{d \times k} (k > \text{polylog}\, d)$ are independent random matrices with independent Gaussian entries, let $M = (A \odot B)^\top (A \odot B) = (A^\top A) * (B^\top B)$, then with high probability*

$$\|M - \text{diag}(M)\| \leq O(d\sqrt{k \log d})$$

Proof: Let $a_1, a_2, ..., a_d \in \mathbb{R}^k$ be the columns of $A^\top$ (the rows of A, but treated as column vectors). We can rewrite $M - \operatorname{diag} M$ as

$$M - \operatorname{diag} M = (\sum_{i\in[d]} a_i a_i^\top) * (B^\top B - \operatorname{diag}(B^\top B)) = \sum_{i\in[d]} (a_i a_i^\top) * (B^\top B - \operatorname{diag}(B^\top B)).$$

Now let $Q = B^\top B - \operatorname{diag}(B^\top B)$, and $M_i = (a_i a_i^\top) * Q$, we would like to bound the spectral norm of the sum $M = \sum_{i\in[d]} M_i$. Clearly these entries are independent, $\mathbb{E}[M_i] = \mathbb{E}[a_i a_i^\top] * Q = I * Q = 0$, so we can apply Matrix Bernstein bound.

Note that when $d < k$, by standard random matrix theory we know $\|Q\| \le O(k)$. Also, every row of Q has norm smaller than the corresponding row of $B^\top B$, which is bounded by $\|B\|\|b_{(i)}\| \le O(\sqrt{kd})$. When $d \ge k$, again by matrix concentration we know $\|Q\| \le O(\sqrt{dk \log d})$. Every row of Q has norm bounded by $O(\sqrt{kd})$ (because entries in a row are independently random, with variance equal to d).

First let us bound the spectral norm for each of the M_i's. Notice that for any vector v, $v^\top[(a_i a_i^\top) * Q]v = (v * a_i)^\top Q(v * a_i)$ by definition of Hadamard product. On the other hand, $\|v * a_i\| \le \|v\|\|a_i\|_\infty$. With high probability $\|a_i\|_\infty \le O(\sqrt{\log k})$, hence $\|M_i\| \le \|a_i\|_\infty^2 \|Q\|$. This is bounded by $O(k \log d)$ when $d < k$ and $O(\sqrt{kd} \log^2 d)$ when $k \le d$.

Next we bound the variance $\|\mathbb{E}[\sum_{i\in[d]} M_i^2]\|$. Since all the M_i's are i.i.d., it suffices to analyze $\mathbb{E}[M_1^2]$. Let $T = \mathbb{E}[M_1^2] = \mathbb{E}[((a_1 a_1^\top) * Q)^2]$, by definition of Hadamard product, we know

$$T_{p,q} = \mathbb{E}[\sum_{r\in[k]} Q_{p,r} Q_{r,q} a_1(p) a_1(q) a_1(r)^2].$$

This number is 0 when $p \ne q$ by independence of entries of a_1. When $p = q$, this is bounded by $3\sum_{r\in[k]} Q_{p,r}^2$ because $\mathbb{E}[a_1(p)^2 a_1(r)^2]$ is 1 when $p \ne r$ and 3 when $p = r$. Therefore $T_{p,p} \le 3\sum_{r\in[k]} Q_{p,r}^2 = 3\|Q^{(p)}\|^2 \le O(dk)$. Since T is a diagonal matrix, we know $\|T\| \le O(dk)$, and $\sigma^2 = \|dT\| = O(d^2 k)$.

By Matrix Bernstein we know with high probability $\|M\| \le O(d\sqrt{k \log d})$. □

Using this lemma, it is easy to get a bound when columns of A, B are unit vectors. In this case, we just need to normalize the columns, the normalization factor is bounded between $d^2/2$ and $2d^2$ with high probability, and therefore, $\|(A^\top A)(B^\top B) - I\| \leq O(\sqrt{k \log d}/d)$.

A.3 Proof of Convergence Results in Theorems 2.4 and 2.5

The main part of the proof is to show that error contraction happens in each iteration of Algorithms 1 and 4 as the two main parts of the algorithm. Then, the contraction result after t iterations is directly argued.

In the following, we first provide a local contraction result for the tensor power iteration (2.6) in Algorithm 1 given noisy tensor $\widehat{T}$. This leads to Lemma 2.2 which is the local convergence guarantee of the tensor power updates. Then, we provide a local contraction argument for the coordinate descent step (2.10) in Algorithm 4.

Combining the above convergence arguments for both updates conclude the overall local convergence guarantee in Theorem. 2.4. Then, combing this local convergence guarantee and the initialization result in Theorem A.1 leads to the global convergence guarantee in Theorem 2.5. In addition, the result in Corollary 2.2 is similarly argued where the bound on the spectral norm of the tensor is argued in Lemma A.5.

A.3.1 Convergence of tensor power iteration: Algorithm 1

In this section, we prove Lemma 2.2 which is the local convergence guarantee of the tensor power updates in Algorithm 1.

Define function $f(\epsilon; k, d)$ as

$$f(\epsilon; k, d) := \alpha \frac{\sqrt{k}}{d} + \frac{2\alpha}{\sqrt{d}} \left(1 + \alpha_0 \sqrt{\frac{k}{d}}\right)^2 \epsilon + \alpha_0 \epsilon^2, \tag{A.5}$$

where $\alpha = \text{polylog}(d)$ and $\alpha_0 = O(1)$. Notice that this function is a small constant when $k < d^{1.5}/\operatorname{poly}\log d$.

Lemma A.9 (Contraction result of Algorithm 1 in one update)**.** *Consider $\widehat{T} = T + \Psi$ as the input to Algorithm 1, where T is a rank-k tensor, and Ψ is a perturbation tensor. Suppose Assumptions (A1)-(A5) hold, and estimates $\widehat{a}$ and $\widehat{b}$ satisfy distance bounds*

$$\begin{aligned}\operatorname{dist}(\widehat{a}, a_j) &\le \epsilon_a,\\ \operatorname{dist}(\widehat{b}, b_j) &\le \epsilon_b,\end{aligned}$$

for some $j \in [k]$, and $\epsilon_a, \epsilon_b > 0$. Let $\epsilon := \max\{\epsilon_a, \epsilon_b\}$, and suppose ψ defined in (A.3) *be small enough such that*[3]

$$w_j - w_j\epsilon^2 - w_{\max} f(\epsilon; k, d) - \psi > 0,$$

where $f(\epsilon; k, d)$ is defined in (A.5)*. Then, update $\widehat{c}$ in* (2.6) *satisfies the following distance bound with high probability (w.h.p.)*

$$\operatorname{dist}(\widehat{c}, c_j) \le \frac{w_{\max} f(\epsilon; k, d) + \psi}{w_j - w_j\epsilon^2 - w_{\max} f(\epsilon; k, d) - \psi}. \tag{A.6}$$

Furthermore, if the bound in (A.6) *is such that $\operatorname{dist}(\widehat{c}, c_j) \le \epsilon$, then the update $\widehat{w} := \widehat{T}(\widehat{a}, \widehat{b}, \widehat{c})$ in* (2.8) *also satisfies w.h.p.*

$$|\widehat{w} - w_j| \le 2w_j\epsilon^2 + w_{\max} f(\epsilon; k, d) + \psi.$$

Remark 27. In the asymptotic regime, $f(\epsilon; k, d)$ is

$$f(\epsilon; k, d) = \tilde{O}\left(\frac{\sqrt{k}}{d}\right) + \tilde{O}\left(\max\left\{\frac{1}{\sqrt{d}}, \frac{k}{d^{3/2}}\right\}\right)\epsilon + O(1)\epsilon^2.$$

Note that the last term is the only effective contracting term. The other terms include a constant term, and the term involving ϵ disappears in only one iteration as long as $k, d \to \infty$, and $\tilde{O}\left(\frac{k}{d^{3/2}}\right) \to 0$.

[3]This is the denominator of bound provided in (A.6).

Remark 28 (Rate of convergence). The local convergence result provided in Theorem 2.4 has a linear convergence rate. But, Algorithm 1 actually provides an almost-quadratic convergence rate in the beginning, and linear convergence rate later on. It can be seen by referring to one-step contraction argument provided in Lemma A.9 where the quadratic term $\alpha_0\epsilon^2$ exists. In the beginning, this term is dominant over linear term involving ϵ, and we have almost-quadratic convergence. Writing $\alpha_0\epsilon^2 = \alpha_0\epsilon^{\zeta}\epsilon^{2-\zeta}$, we observe that we get rate of convergence equal to $2-\zeta$ as long as we have initialization error bounded as $\epsilon_0^{\zeta} = O(1)$. Therefore, we can get arbitrarily close to quadratic convergence with appropriate initialization error. Note that when the model is more overcomplete, the algorithm more rapidly reaches to the linear convergence phase. For the sake of clarity, in proposing Theorem 2.4, we approximated the almost-quadratic convergence rate in the beginning with linear convergence.

Lemma A.9 is proposed in the general form. In Lemma A.10, we provide explicit contraction result by imposing additional perturbation, contraction and initialization Assumptions (A6), (A8) and (A9). We observe that under reasonable rank, perturbation and initialization conditions, the denominator in (A.6) can be lower bounded by a constant, and the numerator is explicitly bounded by a term involving ϵ, and a constant non-contracting term.

Lemma A.10 (Contraction result of Algorithm 1 in one update)**.** *Consider $\widehat{T} = T + \Psi$ as the input to Algorithm 1, where T is a rank-k tensor, and Ψ is a perturbation tensor. Let Assumptions*[4] *(A1)-(A9) hold. Note that initialization bound in (A9) is satisfied for some $j \in [k]$. Then, update $\widehat{c}$ in* (2.6) *satisfies the following distance bound with high probability (w.h.p.)*

$$\text{dist}(\widehat{c}, c_j) \leq \underbrace{\text{Const.}}_{\text{non-contracting term}} + \underbrace{q\epsilon_0}_{\text{contracting term}},$$

where

$$\text{Const.} := \frac{2}{w_{\min}}\left(\psi + w_{\max}\alpha\frac{\sqrt{k}}{d}\right), \tag{A.7}$$

[4] As mentioned in the assumptions, from perturbation bound in (A6), only the bound $\psi \leq \frac{w_{\min}}{6}$ is required here.

and contraction ratio $q < 1/2$ *is defined in* (A.4). *Note that* $\alpha = \text{polylog}(d)$. *In addition, if the above bound be such that* $\text{dist}(\widehat{c}, c_j) \leq \epsilon_0$, *then the update* $\widehat{w} := \widehat{T}(\widehat{a}, \widehat{b}, \widehat{c})$ *in* (2.8) *also satisfies w.h.p.*

$$|\widehat{w} - w_j| \leq \frac{w_{\min}}{2} \text{Const.} + w_{\min} q \epsilon_0.$$

Proof of Lemma 2.2: We incorporate condition (A7) to show that $q < 1/2$ in assumption (A8) is satisfied. In addition, (A7) implies that the bound on ϵ_0 in assumption (A9) holds where it can be shown that the bound in (A9) is bounded as $O(1/\gamma)$. Then, the result is directly proved by iteratively applying the result of Lemma A.10. □

Proof of auxiliary lemmata: tensor power iteration in Algorithm 1

Before providing the proofs, we remind a few definitions and notations.

In Assumption (A2), matrices J_A, J_B, and J_C, are defined as incoherence matrices with zero diagonal entries such that $A^\top A = I + J_A$, $B^\top B = I + J_B$, and $C^\top C = I + J_C$. We have $\max\{\|J_A\|_\infty, \|J_B\|_\infty, \|J_C\|_\infty\} \leq \rho$ as in (A.2).

Given matrix $A \in \mathbb{R}^{d \times k}$, the following notations are defined to refer to its sub-matrices. A_j denotes the j-th column and A^j denotes the j-th row of A. Hence, we have $A_j = a_j, j \in [k]$. In addition, $A_{\setminus j} \in \mathbb{R}^{d \times (k-1)}$ is A with its j-th column removed, and $A^{\setminus j} \in \mathbb{R}^{(d-1) \times k}$ is A with its j-th row removed.

Proof of Lemma A.9: Let $z_a^* \perp a_j$ and $z_b^* \perp b_j$ denote the vectors that achieve supremum value in (2.13) corresponding to $\text{dist}(\widehat{a}, a_j)$ and $\text{dist}(\widehat{b}, b_j)$, respectively. Furthermore, without loss of generality, assume $\|z_a^*\| = \|z_b^*\| = 1$. Then, $\widehat{a}$ and $\widehat{b}$ are decomposed as

$$\widehat{a} = \langle a_j, \widehat{a} \rangle a_j + \text{dist}(\widehat{a}, a_j) z_a^*, \tag{A.8a}$$

$$\widehat{b} = \langle b_j, \widehat{b} \rangle b_j + \text{dist}(\widehat{b}, b_j) z_b^*. \tag{A.8b}$$

Let $\overline{C} := C\,\mathrm{Diag}(w)$ denote the unnormalized matrix C, and $\tilde{c} := \widehat{T}(\widehat{a}, \widehat{b}, I)$ denote the unnormalized update in (2.6). The goal is to bound $\mathrm{dist}\left(\tilde{c}, \overline{C}_j\right)$. Consider any $z_c \perp \overline{C}_j$ such that $\|z_c\| = 1$. Then, we have

$$\langle z_c, \tilde{c}\rangle = \widehat{T}(\widehat{a}, \widehat{b}, z_c) = T(\widehat{a}, \widehat{b}, z_c) + \Psi(\widehat{a}, \widehat{b}, z_c).$$

Substituting $\widehat{a}$ and $\widehat{b}$ from (A.8a) and (A.8b), we have

$$\begin{aligned} T(\widehat{a}, \widehat{b}, z_c) = &\underbrace{\langle a_j, \widehat{a}\rangle\langle b_j, \widehat{b}\rangle T(a_j, b_j, z_c)}_{S_1} + \underbrace{\langle a_j, \widehat{a}\rangle\,\mathrm{dist}(\widehat{b}, b_j) T(a_j, z_b^*, z_c)}_{S_2} \\ &+ \underbrace{\mathrm{dist}(\widehat{a}, a_j)\langle b_j, \widehat{b}\rangle T(z_a^*, b_j, z_c)}_{S_3} + \underbrace{\mathrm{dist}(\widehat{a}, a_j)\,\mathrm{dist}(\widehat{b}, b_j) T(z_a^*, z_b^*, z_c)}_{S_4}. \end{aligned}$$

In the following derivations, we repeatedly use the equality that for any $u, v \in \mathbb{R}^d$, we have $T(u, v, I) = \overline{C}(A^\top u * B^\top v)$. For S_1, we have

$$\begin{aligned} S_1 &\le |T(a_j, b_j, z_c)| = |z_c^\top \overline{C}(A^\top a_j * B^\top b_j)| \\ &= \left| z_c^\top \overline{C}\left[e_j + (J_A * J_B)_j\right]\right| \\ &= \left| z_c^\top \overline{C}_{\setminus j} \left(J_A * J_B\right)_j^{\setminus j}\right| \\ &\le w_{\max}\alpha\frac{\sqrt{k}}{d}, \end{aligned}$$

where equalities $A^\top A = I + J_A$ and $B^\top B = I + J_B$ are exploited in the second equality, and the assumption that $z_c \perp \overline{C}_j$ is used in the last equality. The last inequality is from Assumption (A4). For S_2, we have

$$\begin{aligned} S_2 &\le \epsilon_b |T(a_j, z_b^*, z_c)| = \epsilon_b |z_c^\top \overline{C}(A^\top a_j * B^\top z_b^*)| \\ &= \epsilon_b \left| z_c^\top \overline{C}_{\setminus j}\left[(J_A)_j^{\setminus j} * \left(B_{\setminus j}\right)^\top z_b^*\right]\right| \\ &\le \epsilon_b \left\|\overline{C}_{\setminus j}\right\| \cdot \left\|(J_A)_j^{\setminus j}\right\|_\infty \cdot \left\|\left(B_{\setminus j}\right)^\top z_b^*\right\| \\ &\le w_{\max}\frac{\alpha}{\sqrt{d}}\left(1 + \alpha_0\sqrt{\frac{k}{d}}\right)^2 \epsilon_b, \end{aligned}$$

for some $\alpha = \text{polylog}(d)$ and $\alpha_0 = O(1)$. Second inequality is concluded from $\|u * v\| \leq \|u\|_\infty \cdot \|v\|$, and Assumptions (A2) and (A3) are exploited in the last inequality. Similarly, for S_3, we have

$$\begin{aligned} S_3 &\leq \epsilon_a \left| z_c^\top \overline{C}_{\setminus j} \left[(J_B)_j^{\setminus j} * \left(A_{\setminus j}\right)^\top z_a^* \right] \right| \\ &\leq w_{\max} \frac{\alpha}{\sqrt{d}} \left(1 + \alpha_0 \sqrt{\frac{k}{d}}\right)^2 \epsilon_a. \end{aligned}$$

Finally, for S_4, we have

$$S_4 \leq \epsilon_a \epsilon_b |T(z_a^*, z_b^*, z_c)| \leq \epsilon_a \epsilon_b \|T\| \leq w_{\max} \alpha_0 \epsilon_a \epsilon_b,$$

for some $\alpha_0 = O(1)$. The bound on $\|T\|$ is from Assumption (A4). Note that for random components, we showed in Lemma A.4 that this bound holds w.h.p. exploiting Assumption (A5) and results of Guédon and Rudelson [81]. For the error term $\Psi(\widehat{a}, \widehat{b}, z_c)$, we have

$$\Psi(\widehat{a}, \widehat{b}, z_c) \leq \psi,$$

which is concluded from the definition of spectral norm of a tensor. Note that all vectors $\widehat{a}$, $\widehat{b}$, z_c have unit norm.

Let $\epsilon := \max\{\epsilon_a, \epsilon_b\}$. Then, combining all the above bounds, we have w.h.p.

$$\langle z_c, \tilde{c} \rangle \leq w_{\max} f(\epsilon; k, d) + \psi,$$

where $f(\epsilon; k, d)$ is

$$f(\epsilon; k, d) := \alpha \frac{\sqrt{k}}{d} + \frac{2\alpha}{\sqrt{d}} \left(1 + \alpha_0 \sqrt{\frac{k}{d}}\right)^2 \epsilon + \alpha_0 \epsilon^2.$$

For $\tilde{c}$, we have

$$\begin{aligned}\tilde{c} &= T(\widehat{a},\widehat{b},I) + \Psi(\widehat{a},\widehat{b},I) \\ &= \sum_i w_i \langle a_i,\widehat{a}\rangle\langle b_i,\widehat{b}\rangle c_i + \Psi(\widehat{a},\widehat{b},I) \\ &= w_j \langle a_j,\widehat{a}\rangle\langle b_j,\widehat{b}\rangle c_j + \sum_{i\neq j} w_i \langle a_i,\widehat{a}\rangle\langle b_i,\widehat{b}\rangle c_i + \Psi(\widehat{a},\widehat{b},I),\end{aligned}$$

and therefore,

$$\begin{aligned}\|\tilde{c}\| &\geq \left\| w_j \langle a_j,\widehat{a}\rangle\langle b_j,\widehat{b}\rangle c_j \right\| - \left\| \sum_{i\neq j} w_i \langle a_i,\widehat{a}\rangle\langle b_i,\widehat{b}\rangle c_i \right\| - \|\Psi(\widehat{a},\widehat{b},I)\| \\ &\geq w_j - w_j\epsilon^2 - w_{\max} f(\epsilon;k,d) - \psi,\end{aligned}$$

where inequality $\langle a_j,\widehat{a}\rangle\langle b_j,\widehat{b}\rangle \geq 1-\epsilon^2$ is exploited in the last inequality. Hence, as long as this lower bound on $\|\tilde{c}\|$ is positive (small enough ϵ and ψ), we have

$$\operatorname{dist}(\tilde{c},\overline{C}_j) \leq \frac{w_{\max} f(\epsilon;k,d) + \psi}{w_j - w_j\epsilon^2 - w_{\max} f(\epsilon;k,d) - \psi}. \tag{A.9}$$

Since $\operatorname{dist}(\cdot,\cdot)$ function is invariant with respect to norm, we have $\operatorname{dist}(\widehat{c},c_j) = \operatorname{dist}\left(\tilde{c},\overline{C}_j\right)$ which finishes the proof for bounding $\operatorname{dist}(\widehat{c},c_j)$. Note that $\tilde{c} = \|\tilde{c}\|\widehat{c}$, and $\overline{C}_j = w_j c_j$ where $w_j > 0$.

Now, we provide the bound on $|w_j - \widehat{w}|$. As assumed in the lemma, we have distance bounds

$$\max\left\{\operatorname{dist}(\widehat{a},a_j), \operatorname{dist}\left(\widehat{b},b_j\right), \operatorname{dist}(\widehat{c},c_j)\right\} \leq \epsilon.$$

The estimate $\widehat{w} = \widehat{T}(\widehat{a},\widehat{b},\widehat{c})$ proposed in (2.8) can be expanded as

$$\begin{aligned}\widehat{w} &= T(\widehat{a},\widehat{b},\widehat{c}) + \Psi(\widehat{a},\widehat{b},\widehat{c}) \\ &= \sum_i w_i \langle a_i,\widehat{a}\rangle\langle b_i,\widehat{b}\rangle\langle c_i,\widehat{c}\rangle + \Psi(\widehat{a},\widehat{b},\widehat{c}) \\ &= w_j \langle a_j,\widehat{a}\rangle\langle b_j,\widehat{b}\rangle\langle c_j,\widehat{c}\rangle + \sum_{i\neq j} w_i \langle a_i,\widehat{a}\rangle\langle b_i,\widehat{b}\rangle\langle c_i,\widehat{c}\rangle + \Psi(\widehat{a},\widehat{b},\widehat{c}),\end{aligned}$$

and therefore,

$$
\begin{aligned}
|w_j - \widehat{w}| &\leq \left| w_j \left(1 - \langle a_j, \widehat{a}\rangle \langle b_j, \widehat{b}\rangle \langle c_j, \widehat{c}\rangle\right)\right| + \left|\sum_{i\neq j} w_i \langle a_i, \widehat{a}\rangle \langle b_i, \widehat{b}\rangle \langle c_i, \widehat{c}\rangle\right| + \left|\Psi(\widehat{a}, \widehat{b}, \widehat{c})\right| \\
&\leq w_j \left(1 - \left(1-\epsilon^2\right)^{1.5}\right) + w_{\max} f(\epsilon; k, d) + \psi \\
&\leq 2 w_j \epsilon^2 + w_{\max} f(\epsilon; k, d) + \psi,
\end{aligned}
$$

where $\langle a_j, \widehat{a}\rangle \langle b_j, \widehat{b}\rangle \langle c_j, \widehat{c}\rangle \geq \left(1-\epsilon^2\right)^{1.5}$ is exploited in the second inequality. Notice that this argument is similar to the argument provided earlier for lower bounding $\|\tilde{c}\|$.

□

Proof of Lemma A.10: The result is proved by applying Lemma A.9, and incorporating additional conditions (A6), (A8), and (A9). $f(\epsilon_0; k, d)$ in (A.5) can be bounded as

$$
\begin{aligned}
f(\epsilon_0; k, d) &= \alpha \frac{\sqrt{k}}{d} + \frac{2\alpha}{\sqrt{d}} \left(1 + \alpha_0 \sqrt{\frac{k}{d}}\right)^2 \epsilon_0 + \alpha_0 \epsilon_0^2 \\
&\leq \alpha \frac{\sqrt{k}}{d} + \left[\frac{2\alpha}{\sqrt{d}} \left(1 + \alpha_0 \sqrt{\frac{k}{d}}\right)^2 + \beta'\right] \epsilon_0 \\
&= \alpha \frac{\sqrt{k}}{d} + \frac{w_{\min}}{2 w_{\max}} q \epsilon_0,
\end{aligned}
$$

where $\epsilon_0 \leq \frac{\beta'}{\alpha_0}$ from Assumption (A9) is exploited in the inequality. The last equality is concluded from definition of contracting factor q in (A.4). On the other hand, the denominator in (A.6) can be lower bounded as

$$
w_{\min} \left[1 - \frac{w_{\max}}{w_{\min}} \epsilon_0^2 - \frac{w_{\max}}{w_{\min}} f(\epsilon_0; k, d) - \frac{\psi}{w_{\min}}\right] \geq w_{\min} \left[1 - \frac{1}{6} - \frac{1}{6} - \frac{1}{6}\right] = \frac{w_{\min}}{2},
$$

where Assumptions (A9) and (A6) are used in the inequality. Applying Lemma A.9, the result on $\text{dist}(\widehat{c}, c_j)$ is proved.

From Lemma A.9, we also have

$$\begin{aligned}|\widehat{w} - w_j| &\leq 2w_j\epsilon_0^2 + w_{\max} f(\epsilon_0; k, d) + \psi \\ &\leq \frac{w_{\min}}{2}\,\text{Const.} + 2w_j\epsilon_0^2 + \frac{w_{\min}}{2} q\epsilon_0 \\ &\leq \frac{w_{\min}}{2}\,\text{Const.} + w_{\min} q\epsilon_0.\end{aligned}$$

where $\epsilon_0 \leq \frac{w_{\min} q}{4 w_{\max}}$ from Assumption (A9) is used in the last inequality. □

A.3.2 Convergence of removing residual error: Algorithm 4

In this section, we provide convergence of the coordinate descent of Algorithm 4 for removing the residual error. We first provide the following definition.

Definition A.1 ((η_0, η_1)-nice)**.** *Suppose*

$$\max\{\|A\|, \|B\|, \|C\|\} \leq \eta_1 \sqrt{\frac{k}{d}}.$$

Given an approximate solution $\{\widehat{A}, \widehat{B}, \widehat{C}, \widehat{w}\}$, we call it (η_0, η_1)-nice if matrix $\widehat{A}$ (similarly $\widehat{B}$ and $\widehat{C}$) satisfies

$$\begin{aligned}\|\Delta A_i\| := \|\widehat{a}_i - a_i\| \leq \eta_0 \frac{\sqrt{k}}{d}, \quad \forall i \in [k], \\ \|\widehat{A}\| \leq \eta_1 \sqrt{\frac{k}{d}},\end{aligned}$$

and the weights satisfy

$$|\widehat{w}_i - w_i| \leq \eta_0 w_{\max} \frac{\sqrt{k}}{d}.$$

Given above conditions are satisfied, we prove the following guarantees for removing residual error, Algorithm 4.

Lemma A.11 (Local convergence guarantee of the iterations for removing residual error, Algorithm 4)**.** *Consider T as the input to Algorithm 4, where T is a rank-k tensor. Suppose Assumptions*

(A1)-(A5) and (A11) hold (which are satisfied whp when the components are uniformly i.i.d. drawn from unit d-dimensional sphere). Given initial solution $\left\{\widehat{A}^{(0)}, \widehat{B}^{(0)}, \widehat{C}^{(0)}, \widehat{w}^{(0)}\right\}$ *which is* (η_0, η_1)*-nice, all the following iterations of Algorithm 4 are* $(2\eta_0, 3\eta_1)$*-nice. Furthermore, given the exact tensor* T*, the Frobenius norm error* $\max\{\|\Delta A\|_F, \|\Delta B\|_F, \|\Delta C\|_F, \|\Delta w\|/w_{\min}\}$ *shrinks by at least a factor of 2 in every iteration. In addition, if we have a noisy tensor* $\widehat{T} = T + \Psi$ *such that* $\|\Psi\| \leq \psi$*, then*

$$\max\{\|\Delta A^{(t)}\|_F, \|\Delta B^{(t)}\|_F, \|\Delta C^{(t)}\|_F, \|\Delta w^{(t)}\|/w_{\min}\} \leq 2^{-t}\eta_0 \frac{k}{d} + O\left(\frac{\psi\sqrt{k}}{w_{\min}}\right).$$

Proof: iteration for removing residual error in Algorithm 4

We now prove Lemma A.11 as the local convergence guarantee of the iterations for removing residual error, Algorithm 4.

To prove this lemma, we first observe that the algorithm update formula in (2.10) is (before normalization) $w_i\langle a_i, \widehat{a}_i\rangle\langle b_i, \widehat{b}_i\rangle c_i + \epsilon_i$ where

$$\epsilon_i = \sum_{j\neq i}(w_i\langle a_j, \widehat{a}_i\rangle\langle b_j, \widehat{b}_i\rangle c_j - \widehat{w}_i\langle \widehat{a}_i, \widehat{a}_j\rangle\langle \widehat{b}_i, \widehat{b}_j\rangle \widehat{c}_j).$$

In the following lemma, we show that the error terms ϵ_i's are small.

Lemma A.12. *Before normalization* $\tilde{w}_i\tilde{c}_i = w_i\langle a_i, \widehat{a}_i\rangle\langle b_i, \widehat{b}_i\rangle c_i + \epsilon_i$ *where*

$$\sum_{i=1}^{k}\|\epsilon_i\|^2 \leq o(1)(w_{\max}(\|\Delta(A)\|_F^2 + \|\Delta(B)\|_F^2 + \|\Delta(C)\|_F^2) + \|\Delta w\|^2).$$

Proof: By the update formula in (2.10), we know

$$\epsilon_i = \sum_{j\neq i}(w_i\langle a_j, \widehat{a}_i\rangle\langle b_j, \widehat{b}_i\rangle c_j - \widehat{w}_i\langle \widehat{a}_i, \widehat{a}_j\rangle\langle \widehat{b}_i, \widehat{b}_j\rangle \widehat{c}_j).$$

We expand it into several terms as follows.

$$
\begin{aligned}
\epsilon_i &= \sum_{j\neq i}(w_i\langle a_j,\widehat{a}_i\rangle\langle b_j,\widehat{b}_i\rangle c_j - \widehat{w}_i\langle\widehat{a}_i,\widehat{a}_j\rangle\langle\widehat{b}_i,\widehat{b}_j\rangle\widehat{c}_j)\\
&= \sum_{j\neq i}\langle a_i,a_j\rangle\langle b_i,b_j\rangle(w_jc_j-\widehat{w}_j\widehat{c}_j) \quad \text{(type 1)}\\
&\quad + \sum_{j\neq i} w_j\langle a_j,\Delta A_i\rangle\langle b_j,b_i\rangle c_j + \sum_{j\neq i} w_j\langle a_j,a_i\rangle\langle b_j,\Delta B_i\rangle c_j \quad \text{(type 2)}\\
&\quad - \sum_{j\neq i}\widehat{w}_j\langle a_j,a_i\rangle\langle b_j,\Delta B_i\rangle\widehat{c}_j - \sum_{j\neq i}\widehat{w}_j\langle a_j,a_i\rangle\langle\Delta B_j,\widehat{b}_i\rangle\widehat{c}_j\\
&\quad - \sum_{j\neq i}\widehat{w}_j\langle a_j,\Delta A_i\rangle\langle b_j,b_i\rangle\widehat{c}_j - \sum_{j\neq i}\widehat{w}_j\langle\Delta A_j,\widehat{a}_i\rangle\langle b_j,b_i\rangle\widehat{c}_j\\
&\quad + \sum_{j\neq i}\langle a_j,\Delta A_i\rangle\langle b_j,\Delta B_i\rangle c_j \quad \text{(type 3)}\\
&\quad - \sum_{j\neq i}\widehat{w}_j\langle a_j,\Delta A_i\rangle\langle b_j,\Delta B_i\rangle\widehat{c}_j - \sum_{j\neq i}\widehat{w}_j\langle\Delta A_j,\widehat{a}_i\rangle\langle b_j,\Delta B_i\rangle\widehat{c}_j\\
&\quad - \sum_{j\neq i}\widehat{w}_j\langle a_j,\Delta A_i\rangle\langle\Delta B_j,\widehat{b}_i\rangle\widehat{c}_j - \sum_{j\neq i}\widehat{w}_j\langle\Delta A_j,\widehat{a}_i\rangle\langle\Delta B_j,\widehat{b}_i\rangle\widehat{c}_j.
\end{aligned}
$$

The norm of three different types of terms mentioned above are bounded in Section A.3.2, which conclude the desired bound in the lemma. □

We are now ready to prove main Lemma A.11.

Proof of Lemma A.11: Since $\tilde{w}_i$ is the norm of $w_i\langle a_i,\widehat{a}_i\rangle\langle b_i,\widehat{b}_i\rangle c_i+\epsilon_i$, we know

$$|\tilde{w}_i - w_i| \leq \|\epsilon_i\| + w_i(\Theta(\|\Delta A_i\|^2 + \|\Delta B_i\|^2)),$$

and therefore

$$\|\tilde{w}-w\| \leq o(1)(w_{\max}(\|\Delta(A)\|_F + \|\Delta(B)\|_F + \|\Delta(C)\|_F) + \|\Delta w\|).$$

On the other hand, since the coefficient $w_i\langle a_i, \widehat{a}_i\rangle\langle b_i, \widehat{b}_i\rangle$ is at least $1 - o(1)$, we know $\|\tilde{c}_i - c_i\| \leq 4\|\epsilon_i\|/w_{\min}$. This implies

$$\|\tilde{C} - C\|_F \leq o(1)((\|\Delta(A)\|_F + \|\Delta(B)\|_F + \|\Delta(C)\|_F) + \|\Delta w\|/w_{\min}).$$

By Lemma A.13, we know after the projection procedure, we get $\|\widehat{C} - C\|_F \leq 2\|\tilde{C} - C\|_F$. Therefore combining the two steps we know

$$\|\widehat{C} - C\|_F \leq 2\|\tilde{C} - C\|_F \leq o(1)(\|\Delta(A)\|_F + \|\Delta(B)\|_F + \|\Delta(C)\|_F + \|\Delta w\|/w_{\min}).$$

When we have noise, all the ϵ_i's have an additional term $\Psi(\widehat{a}_i, \widehat{b}_i, I)$ which is bounded by ψ, and thus, the second part of the lemma follows directly.

□

Handling Symmetric Tensors: For symmetric tensors we should change the algorithm as computing the following:

$$T(\widehat{a}_i, \widehat{b}_i, I) - \frac{1}{d}\sum_{i=1}^{d} T(e_i, e_i, I) - \sum_{j\neq i} \widehat{w}_j(\langle \widehat{a}_i, \widehat{a}_j\rangle\langle \widehat{b}_i, \widehat{b}_j\rangle - \frac{1}{d})\widehat{c}_j. \tag{A.10}$$

The result of this will be a change in the term of type 1. Now the Q matrix will be $(A \odot A)^T(A \odot A) - (1 - \frac{1}{d})I - \frac{1}{d}J$ which has desired spectral norm for random matrices.

Claims for proving Lemma A.12

The first term deals with the difference between C and $\widehat{C}$.

Claim 1. *We have*

$$\sqrt{\sum_{i=1}^{k} \|\sum_{j\neq i}\langle a_i, a_j\rangle\langle b_i, b_j\rangle(w_i c_i - \widehat{w}_i\widehat{c}_i)\|^2} \leq o(1)(w_{\max}\|\Delta C\|_F + \|\widehat{w} - w\|).$$

Proof: This sum is equal to the Frobenius norm of a matrix $M = QZ$. Here the matrix Q is a matrix such that is equal to $Q = (A \odot B)^\top (A \odot B) - I$:

$$Q_{i,j} = \begin{cases} \langle a_i, a_j \rangle \langle b_i, b_j \rangle, & i \neq j, \\ 0, & i = j, \end{cases}$$

The matrix Z has columns $Z_i = w_i c_i - \widehat{w}_i \widehat{c}_i$. By assumption we know $\|Q\| \leq o(1)$, and $\|Z\|_F \leq w_{\max} \|\Delta C\|_F + \|\widehat{w} - w\|$. Therefore we have

$$\|M\|_F = \|QZ\|_F \leq \|Q\| \|Z\|_F \leq o(1)(w_{\max} \|\Delta C\|_F + \|\widehat{w} - w\|).$$

□

Of course, in the error ϵ_i, we don't have $\sum_{j \neq i} \langle a_i, a_j \rangle \langle b_i, b_j \rangle w_i c_i$, instead we have terms like $\sum_{j \neq i} \langle \widehat{a}_i, a_j \rangle \langle \widehat{b}_i, b_j \rangle w_i c_i$. The next two lemmas show that these two terms are actually very close.

Claim 2. *We have*

$$\sqrt{\sum_{i=1}^k \| \sum_{j \neq i} \langle \Delta A_i, \widehat{a}_j \rangle \langle b_i, b_j \rangle \widehat{w}_i \widehat{c}_i \|^2} \leq o(w_{\max}) \|\Delta A\|_F.$$

$$\sqrt{\sum_{i=1}^k \| \sum_{j \neq i} \langle \Delta A_j, \widehat{a}_i \rangle \langle b_i, b_j \rangle \widehat{w}_i \widehat{c}_i \|^2} \leq o(w_{\max}) \|\Delta A\|_F.$$

Same is true if any $\widehat{\cdot}$ is replaced by the true value.

Proof: Similar as before, we treat the left hand side as the Frobenius norm of some matrix $M = QZ$. Here $Z_i = \widehat{w}_i \widehat{c}_i$, and Q is the following matrix:

$$Q_{i,j} = \begin{cases} \langle \Delta A_i, \widehat{a}_j \rangle \langle b_i, b_j \rangle, & i \neq j, \\ 0, & i = j, \end{cases}$$

We shall bound $\|M\|_F$ by $\|Z\| \|Q\|_F$. By assumption we know $\|Z\| \leq w_{\max} \cdot 2\eta_1 \sqrt{k/d} = O(w_{\max} \sqrt{k/d})$. On the other hand, we know $\langle b_i, b_j \rangle \leq \tilde{O}(1/\sqrt{d})$ hence $\|Q\|_F \leq \tilde{O}(1/\sqrt{d}) \|\widehat{A}^T \Delta A\|_F \leq \tilde{O}(1/\sqrt{d}) \|\widehat{A}\| \|\Delta A\|_F =$

$\tilde{O}(\sqrt{k}/d)\|\Delta A\|_F$. Therefore we have

$$\|M\|_F \leq \|Z\|\|Q\|_F \leq O(w_{\max}\sqrt{k/d})\cdot\tilde{O}(w_{\max}\sqrt{k}/d)\|\Delta A\|_F = \tilde{O}(k/d\sqrt{d})\|\Delta A\|_F = o(w_{\max})\|\Delta A\|_F.$$

Notice that the proof works for both terms. □

Claim 3. *We have*

$$\sqrt{\sum_{i=1}^{k}\|\sum_{j\neq i}\langle\Delta A_i,\widehat{a}_j\rangle\langle\Delta B_i,\widehat{b}_j\rangle\widehat{w}_i\widehat{c}_i\|^2} \leq o(w_{\max})(\|\Delta A\|_F + \|\Delta B\|_F).$$

The same is true if the inner-products are between $\langle\Delta A_j,\widehat{a}_i\rangle$ or $\langle\Delta B_j,\widehat{b}_i\rangle$, or if any $\widehat{\cdot}$ is replaced by the true value.

Proof: Similar as before, we treat the left hand side as the Frobenius norm of some matrix $M = QZ$. Here $Z_i = \widehat{w}_i\widehat{c}_i$, and Q is the following matrix

$$Q_{i,j} = \begin{cases} \langle\Delta A_i,\widehat{a}_j\rangle\langle\Delta B_i,b_j\rangle, & i\neq j, \\ 0, & i=j, \end{cases}$$

Now using definition of $2\to 4$ norm and $2ab \leq a^2+b^2$ we first bound the Frobenius norm of the matrix Q:

$$\sum_{i\neq j}(\langle\Delta A_i,\widehat{a}_j\rangle\langle\Delta B_i,\widehat{b}_j\rangle)^2 \leq \sum_{i\neq j}(\langle\Delta A_i,\widehat{a}_j\rangle)^4+(\langle\Delta B_i,\widehat{b}_j\rangle)^4 \leq \sum_{i=1}^{k}\|\widehat{A}^\top\|_{2\to 4}\|\Delta A_i\|^4+\|\widehat{B}^\top\|_{2\to 4}\|\Delta B_i\|^4$$

Now we first bound the $2\to 4$ norm of the matrix $\widehat{A}^\top = A^\top + \Delta A^\top$. By assumption we already know $\|A^\top\|_{2\to 4} \leq O(1)$. On the other hand, for any unit vector u

$$\sum_{i=1}^{k}\langle\Delta A_i,u\rangle^4 \leq \max_{i=1}^{k}\langle\Delta A_i,u\rangle^2\sum_{i=1}^{k}\langle\Delta A_i,u\rangle^2 \leq \tilde{O}(k^2/d^3) = o(1).$$

Here we used the assumption that $\|\Delta A_i\| \leq \tilde{O}(\sqrt{k}/d)$ and $\|\Delta A\| \leq O(\sqrt{k/d})$. Therefore $\|\widehat{A}^\top\|_{2\to 4} \leq \|A^\top\|_{2\to 4} + \|\Delta A^\top\|_{2\to 4} \leq O(1)$ (and similarly for $\widehat{B}^\top$).

Therefore

$$\begin{aligned}\|Q\|_F &\le \sqrt{\sum_{i=1}^{k} \|\widehat{A}^\top\|_{2\to 4}\|\Delta A_i\|^4 + \|\widehat{B}^\top\|_{2\to 4}\|\Delta B_i\|^4} \\ &\le O(1)\sqrt{\sum_{i=1}^{k} \|\Delta A_i\|^4 + \|\Delta B_i\|^4} \\ &\le O(1)\cdot \max_{i=1}^{k}(\|\Delta A\|_i + \|\Delta B\|_i)\sqrt{\sum_{i=1}^{k} \|\Delta A_i\|^2 + \|\Delta B_i\|^2} \\ &\le \tilde{O}(\sqrt{k}/d)(\|\Delta A\|_F + \|\Delta B\|_F).\end{aligned}$$

On the other hand we know $\|Z\| \le O(w_{\max}\sqrt{k/d})$, hence $\|M\|_F \le \|Z\|\|Q\|_F \le o(w_{\max})(\|\Delta A\|_F + \|\Delta B\|_F)$.

□

Projection Procedure 5

In this section, we describe the functionality of projection Procedure 5. Suppose the initial solution $\{\widehat{A}^0, \widehat{B}^0, \widehat{C}^0, \widehat{w}^0\}$ is (η_0, η_1)-nice. Then, given an arbitrary solution $\{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{w}\}$, we run projection Procedure 5 to get a $(2\eta_0, 4\eta_1)$-nice solution without losing too much in Frobenius norm error. This is shown in the following Lemma.

Lemma A.13. *Suppose the initial solution $\{\widehat{A}^0, \widehat{B}^0, \widehat{C}^0, \widehat{w}^0\}$ is (η_0, η_1)-nice. For any solution $\{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{w}\}$, let error $E = \max\{\|\tilde{A} - A\|_F, \|\tilde{B} - B\|_F, \|\tilde{C} - C\|_F, \|\tilde{w} - w\|/w_{\min}\}$. Then after the projection Procedure 5, the new solution is $(2\eta_0, 3\eta_1)$-nice and has error at most $2E$.*

Proof: Intuitively, by truncating D the matrix we get is closest to $\tilde{A}$ among matrices with spectral norm $\eta_1\sqrt{k/d}$. We first prove this fact:

Claim 4.

$$\|Q - \tilde{A}\|_F = \min_{\|M\| \le \eta_1\sqrt{k/d}} \|M - \tilde{A}\|_F.$$

Proof: By symmetric properties of Frobenius and spectral norm (both are invariant under rotation), we can rotate the matrices $Q, M, \tilde{A}$ simultaneously, so that $\tilde{A}$ becomes a diagonal matrix D. Since M has spectral norm bounded by $\eta_1\sqrt{k/d}$, in particular all its entries must be bounded by $\eta_1\sqrt{k/d}$. Also, we know $\|D - \widehat{D}\|_F = \min_{\forall(i,j)M_{i,j}\leq\eta_1\sqrt{k/d}} \|D - M\|_F$, therefore $\|D - \widehat{D}\|_F = \min_{\|M\|\leq\eta_1\sqrt{k/d}} \|D - M\|_F$. By the rotation invariant property this implies the claim. □

Since the optimal solution A has spectral norm bounded by $\eta_1\sqrt{k/d}$, in particular from above claim we know $\|Q - \tilde{A}\|_F \leq \|\tilde{A} - A\|_F$. By triangle inequality we get $\|Q - A\|_F \leq 2E$. In the next step we are essentially projecting the solution Q to a convex set that contains A (the set of matrices that are column-wise $\eta_1\sqrt{k}/d$ close to $\widehat{A}^0$), so the distance can only decrease. Similar arguments work for $\widehat{B}, \widehat{C}, \widehat{w}$, therefore the error of the new solution is bounded by $2E$.

By construction it is clear that the columns of the new solution is within $\eta_0\sqrt{k}/d$ to the columns of the initial solution, so they must be within $2\eta_0\sqrt{k}/d$ to the columns of the true solution. The only thing left to prove is that $\|\widehat{A}\| \leq 3\eta_1\sqrt{k/d}$.

First we observe that $\widehat{A} = \widehat{A}^0 + Z$ where Z is a matrix whose columns are multiples of $Q - \widehat{A}^0$, and the multiplier is never larger than 1. Therefore $\|\widehat{A}\| \leq \|hA^0\| + \|Z\| \leq \|\widehat{A}^0\| + \|Q - \widehat{A}^0\| \leq 2\|\widehat{A}^0\| + \|Q\| \leq 3\eta_1\sqrt{k/d}$. □

A.4 SVD Initialization Result

In this section, we analyze the SVD-based initialization technique proposed in Procedure 2. The goal is to provide good initialization vectors close to the columns of true components A and B in the regime of $k = O(d)$.

Given a vector $\theta \in \mathbb{R}^d$, matrix $T(I, I, \theta)$ results a linear combination of slices of tensor T. For tensor T in (A.1), we have

$$T(I, I, \theta) = \sum_{i\in[k]} w_i\langle\theta, c_i\rangle a_i b_i^\top = \sum_{i\in[k]} \lambda_i a_i b_i^\top = A\,\mathrm{Diag}(\lambda)B^\top, \tag{A.11}$$

where $\lambda_i := w_i\langle\theta, c_i\rangle, i \in [k]$, and $\lambda := [\lambda_1, \lambda_2, \ldots, \lambda_k]^\top \in \mathbb{R}^k$ is expressed as

$$\lambda = \text{Diag}(w)C^\top\theta.$$

Since A and B are not orthogonal matrices, the expansion in (A.11) is not the SVD [5] of $T(I, I, \theta)$. But, we show in the following theorem that if we draw enough number of random vectors θ in the regime of $k = O(d)$, we can eventually provide good initialization vectors through SVD of $T(I, I, \theta)$. Define

$$g(L) := \sqrt{2\ln(L)} - \frac{\ln(\ln(L)) + c}{2\sqrt{2\ln(L)}} - \sqrt{2\ln(k)}.$$

Theorem A.1 (SVD initialization when $k = O(d)$)**.** *Consider tensor $\widehat{T} = T + \Psi$ where T is a rank-k tensor, and Ψ is a perturbation tensor. Let Assumptions (A1)-(A3) hold and $k = O(d)$. Draw L i.i.d. random vectors $\theta^{(j)} \sim \mathcal{N}(0, I_d), j \in [L]$. Let $u_1^{(j)}$ and $v_1^{(j)}$ be the top left and right singular vectors of $\widehat{T}(I, I, \theta^{(j)})$. This is L random runs of Procedure 2. Suppose L satisfies the bound*

$$g(L) \geq \frac{w_{\max}(1+\mu)}{w_{\min} - \rho w_{\max}(1+\mu)} 4\sqrt{\log k},$$

with $\mu = \frac{2\mu_R + \tilde{\mu} - 1}{1 - \tilde{\mu}} < \frac{w_{\min}}{w_{\max}\rho} - 1$, for μ_R and $\mu_{\min}$ defined in (A.14), and some $0 < \tilde{\mu} < 1$. Note that $\rho \leq \frac{\alpha}{\sqrt{d}}$ is also defined as the incoherence parameter in Assumption (A2). Then, w.h.p., at least one of the pairs $(u_1^{(j)}, v_1^{(j)}), j \in [L]$, say j^*, satisfies

$$\max\left\{\text{dist}\left(u_1^{(j^*)}, a_1\right), \text{dist}\left(v_1^{(j^*)}, b_1\right)\right\} \leq \frac{4w_{\max}\mu_{\min}(1+\rho)\sqrt{\log k} + \alpha_0\sqrt{d}\psi}{w_{\min}\tilde{\mu}g(L) - \alpha_0\sqrt{d}\psi},$$

where $\psi := \|\Psi\|$ is the spectral norm of perturbation tensor Ψ, and $\alpha_0 > 1$ is a constant.

Proof: Let $\lambda^{(j)} := \text{Diag}(w)C^\top\theta^{(j)} \in \mathbb{R}^k$ and $\tilde{\lambda}^{(j)} := C^\top\theta^{(j)} \in \mathbb{R}^k$. From Lemmata A.14 and A.15, there exists a $j^* \in [L]$ such that w.h.p., we have

$$\max\left\{\text{dist}\left(u_1^{(j^*)}, a_1\right), \text{dist}\left(v_1^{(j^*)}, b_1\right)\right\} \leq \frac{\mu_{\min}\lambda_{(2)} + \|\Psi(I, I, \theta)\|}{\tilde{\mu}\lambda_1 - \|\Psi(I, I, \theta)\|}.$$

[5] Note that if A and B are orthogonal matrices, columns of A and B are directly recovered by computing SVD of $T(I, I, \theta)$.

From (A.12), with probability at least $1 - 2k^{-1}$, we have

$$\lambda_1^{(j^*)} \geq w_{\min} g(L).$$

From (A.13), with probability at least $1 - k^{-7}$, we have

$$\lambda_{(2)}^{(j^*)} \leq w_{\max} \left(\rho \tilde{\lambda}_1^{(j^*)} + 4\sqrt{\log k} \right) \leq 4 w_{\max} (1+\rho) \sqrt{\log k},$$

where in the last inequality, we also applied upper bound on $\tilde{\lambda}_1^{(j^*)}$. Combining all above bounds and Lemma A.19 finishes the proof. □

A.4.1 Auxiliary lemmata for initialization

In the following Lemma, we show that the gap condition between the maximum and the second maximum of vector λ required in Lemma A.15 is satisfied under some number of random draws.

Lemma A.14 (Gap condition)**.** *Consider an arbitrary matrix $C \in \mathbb{R}^{d \times k}$ with unit-norm columns which also satisfies incoherence condition $\max_{i \neq j} |\langle c_i, c_j \rangle| \leq \rho$ for some $\rho > 0$. Let*

$$\lambda := \mathrm{Diag}(w) C^\top \theta \in \mathbb{R}^k,$$

denote the vector that captures correlation of $\theta \in \mathbb{R}^d$ with columns of C. Without loss of generality, assume that $\lambda_1 = \max_i |\lambda_i|$, and let $\lambda_{(2)} := \max_{i \neq 1} |\lambda_i|$. Draw L i.i.d. random vectors $\theta^{(j)} \sim \mathcal{N}(0, I_d), j \in [L]$, and $\lambda^{(j)} := \mathrm{Diag}(w) C^\top \theta^{(j)}$. Suppose L satisfies the bound

$$\sqrt{\frac{\ln(L)}{8\ln(k)}} \left(1 - \frac{\ln(\ln(L)) + c}{4\ln(L)} - \sqrt{\frac{\ln(k)}{\ln(L)}} \right) \geq \frac{w_{\max}(1+\mu)}{w_{\min} - \rho w_{\max}(1+\mu)},$$

for some $0 < \mu < \frac{w_{\min}}{w_{\max}\rho} - 1$. Then, with probability at least $1 - 2k^{-1} - k^{-7}$, we have the following gap condition for at least one draw, say j^*,

$$\lambda_1^{(j^*)} \geq (1+\mu) \lambda_{(2)}^{(j^*)}.$$

Proof: Define $\tilde{\lambda} := \text{Diag}(w)^{-1}\lambda = C^\top\theta$. We have $\lambda_j = w_j\tilde{\lambda}_j, j \in [k]$.

Each vector $\tilde{\lambda}^{(j)}$ is a random Gaussian vector $\tilde{\lambda}^{(j)} \sim \mathcal{N}(0, C^\top C)$. Let $j^* := \arg\max_{j\in[L]} \tilde{\lambda}_1^{(j)}$. Since $\max_{j\in[L]} \tilde{\lambda}_1^{(j)}$, is a 1-Lipschitz function of L independent $\mathcal{N}(0,1)$ random variables, similar to the analysis in Lemma B.1 of Anandkumar et al. [15], we have

$$\Pr\left[\tilde{\lambda}_1^{(j^*)} \geq \sqrt{2\ln(L)} - \frac{\ln(\ln(L)) + c}{2\sqrt{2\ln(L)}} - \sqrt{2\ln(k)}\right] \geq 1 - \frac{2}{k}. \tag{A.12}$$

Any vector $c_i, i \neq 1$, can be decomposed to two components parallel and perpendicular to c_1 as $c_i = \langle c_i, c_1\rangle c_1 + \mathcal{P}_{\perp_{c_1}}(c_i)$. Then, for any $\tilde{\lambda}_i, i \neq 1$, we have

$$\tilde{\lambda}_i := \langle\theta, c_i\rangle = \underbrace{\theta^\top\langle c_i, c_1\rangle c_1}_{=:\tilde{\lambda}_{i,\parallel}} + \underbrace{\theta^\top\mathcal{P}_{\perp_{c_1}}(c_i)}_{=:\tilde{\lambda}_{i,\perp}}.$$

Since $\mathcal{P}_{\perp_{c_1}}(c_i) \perp c_1, i \neq 1$, we have $\tilde{\lambda}_{i,\perp}, i \neq 1$, are independent of $\tilde{\lambda}_1 := \theta^\top c_1$, and therefore, the following bound can be argued independent of bound in (A.12). From Lemma A.17, we have

$$\Pr\left[\max_{i\neq 1} \tilde{\lambda}_{i,\perp}^{(j^*)} \geq 4\sqrt{\log k}\right] \leq k^{-7}.$$

For $\tilde{\lambda}_{i,\parallel}$, we have

$$\tilde{\lambda}_{i,\parallel} = \theta^\top\langle c_i, c_1\rangle c_1 \leq \rho\theta^\top c_1 = \rho\tilde{\lambda}_1,$$

where we also assumed that $\tilde{\lambda}_1 := \theta^\top c_1 > 0$ which is true for large enough L, concluded from (A.12). By combining above two bounds, with probability at least $1 - k^{-7}$, we have

$$\tilde{\lambda}_{(2)}^{(j^*)} \leq \rho\tilde{\lambda}_1 + 4\sqrt{\log k}. \tag{A.13}$$

From the given bound on L in the lemma and inequalities (A.12) and (A.13), with probability at least $1 - 2k^{-1} - k^{-7}$, we have

$$\tilde{\lambda}_1^{(j^*)} \geq \frac{w_{\max}(1+\mu)}{w_{\min} - \rho w_{\max}(1+\mu)}\left(\tilde{\lambda}_{(2)}^{(j^*)} - \rho\tilde{\lambda}_1^{(j^*)}\right).$$

Simple calculations imply that

$$w_{\min}\tilde{\lambda}_1^{(j^*)} \geq (1+\mu) w_{\max}\tilde{\lambda}_{(2)}^{(j^*)}.$$

Incorporating inequalities $\lambda_1 \geq w_{\min}\tilde{\lambda}_1$ and $\lambda_{(2)} \leq w_{\max}\tilde{\lambda}_{(2)}$ finishes the proof saying that the result of lemma is valid for the j^*-th draw. □

In the following lemma, we show that if a vector $\theta \in \mathbb{R}^d$ is relatively more correlated with c_1 (comparing to $c_i, i \neq 1$), then dominant singular vectors of $\widehat{T}(I, I, \theta)$ provide good initialization vectors for a_1 and b_1.

Before proposing the lemma, we define

$$\mu_E := \alpha\sqrt{\frac{k}{d}}\left(2 + 2\alpha_0\sqrt{\frac{k}{d}} + \frac{\alpha}{\sqrt{d}}\right), \quad \mu_R := \left(1 + \alpha_0\sqrt{\frac{k}{d}}\right)^2, \quad \mu_{\min} := \min\{\mu_E, \mu_R\}. \tag{A.14}$$

where $\alpha = \text{polylog}(d)$, and $\alpha_0 > 0$ is a constant.

Lemma A.15. *Consider $\widehat{T} = T + \Psi$, where T is a rank-k tensor, and Ψ is a perturbation tensor. Let assumptions (A1)-(A3) hold for T. Let u_1 and v_1 be the top left and right singular vectors of $\widehat{T}(I, I, \theta)$. Let*

$$\lambda := \text{Diag}(w)C^\top\theta \in \mathbb{R}^k,$$

denote the vector that captures correlation of θ with different $c_i, i \in [k]$, weighted by $w_i, i \in [k]$. Without loss of generality, assume that $\lambda_1 = \max_i |\lambda_i|$, and let $\lambda_{(2)} := \max_{i \neq 1} |\lambda_i|$. Suppose the relative gap condition

$$\lambda_1 \geq (1+\mu)\lambda_{(2)}, \tag{A.15}$$

is satisfied for some $\mu > \frac{\lambda_1}{\lambda_1 - \|\Psi(I,I,\theta)\|} 2\mu_R - 1$, where μ_R and $\mu_{\min}$ are defined in (A.14)*. Then, with high probability (w.h.p.),*

$$\max\{\text{dist}(u_1, a_1), \text{dist}(v_1, b_1)\} \leq \frac{\mu_{\min}\lambda_{(2)} + \|\Psi(I,I,\theta)\|}{\tilde{\mu}\lambda_1 - \|\Psi(I,I,\theta)\|},$$

for $\|\Psi(I,I,\theta)\|/\lambda_1 < \tilde{\mu} < 1$ *defined as*

$$\tilde{\mu} := \frac{1+\mu-2\mu_R}{1+\mu}.$$

Proof: From Assumption (A1), $T(I,I,\theta)$ can be written as equation (A.11), Expanded as

$$T(I,I,\theta) = \lambda_1 a_1 b_1^\top + \underbrace{\sum_{i\neq 1} \lambda_i a_i b_i^\top}_{=:R}.$$

From here, we prove the result in two cases. First when $\mu_E < \mu_R$ and therefore $\mu_{\min} = \mu_E$, and second when $\mu_E \geq \mu_R$ and therefore $\mu_{\min} = \mu_R$.

Case 1 ($\mu_E < \mu_R$): According to the subspaces spanned by a_1 and b_1, we decompose matrix R to two components as $R = \mathcal{P}_\perp(R) + \mathcal{P}_\parallel(R)$. First term $\mathcal{P}_\perp(R)$ is the component with column space orthogonal to a_1 and row space orthogonal to b_1, and $\mathcal{P}_\parallel(R)$ is the component with either the column space equal to a_1 or the row space equal to b_1. We have

$$\begin{aligned}
\mathcal{P}_\perp(R) &= (I - P_{a_1})R(I - P_{b_1}),\\
\mathcal{P}_\parallel(R) &= P_{a_1}R + RP_{b_1} - P_{a_1}RP_{b_1},
\end{aligned}$$

where $P_{a_1} = a_1 a_1^\top$ is the projection operator on the subspace in $\mathbb{R}^d$ spanned by a_1, and similarly $P_{b_1} = b_1 b_1^\top$ is the projection operator on the subspace in $\mathbb{R}^d$ spanned by b_1. Thus, for $\widehat{T} = T + \Psi$, we have

$$\widehat{T}(I,I,\theta) = \underbrace{\lambda_1 a_1 b_1^\top + \mathcal{P}_\perp(R)}_{=:M} + \underbrace{\mathcal{P}_\parallel(R)}_{=:E} + \Psi(I,I,\theta).$$

Looking at M, it becomes more clear why we proposed the above decomposition for R. Since the column and row space of $\mathcal{P}_\perp(R)$ are orthogonal to a_1 and b_1, respectively, the SVD of M has a_1 and b_1 as its left and right singular vectors, respectively. Hence, M has the SVD form

$$M = [a_1\ \tilde{U}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \tilde{\Sigma}_2 \end{bmatrix} [b_1\ \tilde{V}_2]^\top,$$

where $\mathcal{P}_\perp(R) = \tilde{U}_2\tilde{\Sigma}_2\tilde{V}_2^\top$ is the SVD of $\mathcal{P}_\perp(R)$. Let $\tilde{\sigma}_2 := \max_i(\tilde{\Sigma}_2)_{ii}$. From gap condition (A.15) assumed in the lemma and inequality (A.16), we have $\lambda_1 \geq \tilde{\sigma}_2$, and therefore, a_1 and b_1 are the top left and right singular vectors of M. On the other hand, $\widehat{T}(I, I, \theta)$ has the corresponding SVD form

$$\widehat{T}(I, I, \theta) = [u_1 \; U_2] \begin{bmatrix} \sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} [v_1 \; V_2]^\top,$$

where u_1 and v_1 are its top left and right singular vectors. We have

$$\begin{aligned} \tilde{\sigma}_2 = \|\mathcal{P}_\perp(R)\| &\leq \|R\| \\ &= \left\| \sum_{i=2}^{k} \lambda_i a_i b_i^\top \right\| \\ &\leq \lambda_{(2)} \left\| A_{\setminus 1} \right\| \left\| B_{\setminus 1}^\top \right\| \\ &\leq \lambda_{(2)} \|A\| \left\| B^\top \right\| \\ &\leq \left(1 + \alpha_0 \sqrt{\frac{k}{d}}\right)^2 \lambda_{(2)} =: \mu_R \lambda_{(2)}, \end{aligned} \tag{A.16}$$

where the sub-multiplicative property of spectral norm is used in the second inequality, and the last inequality is from Assumption (A3). From Weyl's theorem, we have

$$\begin{aligned} |\sigma_1 - \lambda_1| &\leq \|E\| + \|\Psi(I, I, \theta)\| \\ &\leq \lambda_{(2)} \alpha \sqrt{\frac{k}{d}} \left(2 + 2\alpha_0 \sqrt{\frac{k}{d}} + \frac{\alpha}{\sqrt{d}}\right) + \|\Psi(I, I, \theta)\| \\ &=: \mu_E \lambda_{(2)} + \|\Psi(I, I, \theta)\|, \end{aligned} \tag{A.17}$$

where (A.18) is used in the second inequality. Therefore, we have

$$\begin{aligned}
\sigma_1 - \tilde{\sigma}_2 &= \sigma_1 - \lambda_1 + \lambda_1 - \tilde{\sigma}_2 \\
&\geq -\mu_E \lambda_{(2)} - \|\Psi(I,I,\theta)\| + \lambda_1 - \mu_R \lambda_{(2)} \\
&\geq \left(1 - \frac{\mu_E + \mu_R}{1+\mu}\right)\lambda_1 - \|\Psi(I,I,\theta)\|, \\
&=: \tilde{\mu}_1 \lambda_1 - \|\Psi(I,I,\theta)\| =: \nu,
\end{aligned}$$

where bounds (A.16) and (A.17) are used in the first inequality, and the second inequality is concluded from the gap condition (A.15) assumed in the lemma. Therefore, since $\sigma_1 \geq \beta + \nu$ and $\tilde{\sigma}_2 \leq \beta$ for some $\beta > 0$, Wedin's theorem is applied to the equality $\widehat{T}(I,I,\theta) = M + E + \Psi(I,I,\theta)$, which implies that

$$\begin{aligned}
\max\left\{\sqrt{1-\langle u_1, a_1\rangle^2}, \sqrt{1-\langle v_1, b_1\rangle^2}\right\} &\leq \frac{\|E + \Psi(I,I,\theta)\|}{\nu} \\
&\leq \frac{\mu_E \lambda_{(2)} + \|\Psi(I,I,\theta)\|}{\tilde{\mu}_1 \lambda_1 - \|\Psi(I,I,\theta)\|} \\
&\leq \frac{\mu_{\min} \lambda_{(2)} + \|\Psi(I,I,\theta)\|}{\tilde{\mu} \lambda_1 - \|\Psi(I,I,\theta)\|},
\end{aligned}$$

where we used $\mu_{\min} = \mu_E$ and $\tilde{\mu}_1 > \tilde{\mu}$ in the last inequality when $\mu_E < \mu_R$. Since $\mathrm{dist}^2(u_1, a_1) + \langle u_1, a_1\rangle^2 = 1$, the proof is complete for this case.

Bounding the spectral norm of E: For any $i \neq j$, let $\rho_{ij}^{(a)} := |\langle a_i, a_j\rangle|$ and $\rho_{ij}^{(b)} := |\langle b_i, b_j\rangle|$. We have

$$\begin{aligned}
E := \mathcal{P}_{\|}(R) &= P_{a_1} R + R P_{b_1} - P_{a_1} R P_{b_1}, \\
&= a_1 a_1^\top R + R b_1 b_1^\top - a_1 a_1^\top R b_1 b_1^\top \\
&= \sum_{i \neq 1} \lambda_i a_1 a_1^\top a_i b_i^\top + \sum_{i \neq 1} \lambda_i a_i b_i^\top b_1 b_1^\top - \sum_{i \neq 1} \lambda_i a_1 a_1^\top a_i b_i^\top b_1 b_1^\top \\
&= \sum_{i \neq 1} \lambda_i \rho_{1i}^{(a)} a_1 b_i^\top + \sum_{i \neq 1} \lambda_i \rho_{1i}^{(b)} a_i b_1^\top - \sum_{i \neq 1} \lambda_i \rho_{1i}^{(a)} \rho_{1i}^{(b)} a_1 b_1^\top \\
&= \underbrace{A_{(1)} \operatorname{Diag}(\lambda_{(a)}) B_{\setminus 1}^\top}_{E_1} + \underbrace{A_{\setminus 1} \operatorname{Diag}(\lambda_{(b)}) B_{(1)}^\top}_{E_2} - \underbrace{A_{(1)} \operatorname{Diag}(\lambda_{(a,b)}) B_{(1)}^\top}_{E_3},
\end{aligned}$$

where $A_{(1)} := \left[\overbrace{a_1|a_1|\cdots|a_1}^{k-1 \text{ times}}\right] \in \mathbb{R}^{d\times(k-1)}$, $B_{\setminus 1} := [b_2|b_3|\cdots|b_k] \in \mathbb{R}^{d\times(k-1)}$, and $\lambda_{(a)} := [\lambda_i \rho_{1i}^{(a)}]_{i\neq 1} \in \mathbb{R}^{k-1}$. The other notations are similarly defined.

For E_1, we have

$$\begin{aligned}
\|E_1\| &\leq \|A_{(1)}\,\mathrm{Diag}(\lambda_{(a)})\|\|B_{\setminus 1}^\top\| \\
&= \|\lambda_{(a)}\|\|a_1\|\|B_{\setminus 1}^\top\| \\
&\leq \sqrt{k}\lambda_{(2)}\rho\|B^\top\| \\
&\leq \lambda_{(2)}\alpha\sqrt{\frac{k}{d}}\left(1+\alpha_0\sqrt{\frac{k}{d}}\right).
\end{aligned}$$

Where the first equality is concluded from Lemma A.18, and Assumptions (A2) and (A3) are exploited in the last inequality. Similarly, for E_2 and E_3, we have

$$\begin{aligned}
\|E_2\| &\leq \lambda_{(2)}\alpha\sqrt{\frac{k}{d}}\left(1+\alpha_0\sqrt{\frac{k}{d}}\right), \\
\|E_3\| &\leq \lambda_{(2)}\alpha^2\frac{\sqrt{k}}{d}.
\end{aligned}$$

Therefore, we have

$$\|E\| \leq \lambda_{(2)}\alpha\sqrt{\frac{k}{d}}\left(2+2\alpha_0\sqrt{\frac{k}{d}}+\frac{\alpha}{\sqrt{d}}\right). \tag{A.18}$$

Case 2 ($\mu_R \leq \mu_E$): The result can be similarly achieved when $\mu_R \leq \mu_E$. Here we directly apply Wedin's theorem to $\widehat{T}(I,I,\theta) = \lambda_1 a_1 b_1^\top + R + \Psi(I,I,\theta)$, treating $R+\Psi(I,I,\theta)$ as the error term. From Weyl's theorem, we have

$$\sigma_1 \geq \lambda_1 - \|R\| - \|\Psi(I,I,\theta)\| \geq \underbrace{\left(1-\frac{\mu_R}{1+\mu}\right)}_{=:\tilde{\mu}_2}\lambda_1 - \|\Psi(I,I,\theta)\|,$$

where (A.16) and gap condition (A.15) are used in the second inequality. Since $\tilde{\sigma}_2 = 0$, by Wedin's theorem, we have

$$\begin{aligned}\max\left\{\sqrt{1-\langle u_1, a_1\rangle^2}, \sqrt{1-\langle v_1, b_1\rangle^2}\right\} &\leq \frac{\mu_R\lambda_{(2)} + \|\Psi(I,I,\theta)\|}{\tilde{\mu}_2\lambda_1 - \|\Psi(I,I,\theta)\|} \\ &\leq \frac{\mu_{\min}\lambda_{(2)} + \|\Psi(I,I,\theta)\|}{\tilde{\mu}\lambda_1 - \|\Psi(I,I,\theta)\|},\end{aligned}$$

where we used $\mu_{\min} = \mu_R$ and $\tilde{\mu}_2 \geq \tilde{\mu}$ in the last inequality when $\mu_R \leq \mu_E$. Since $\text{dist}^2(u_1, a_1) + \langle u_1, a_1\rangle^2 = 1$, the proof is complete for this case. □

The above lemma concludes the proof for initialization procedure, except for a few auxiliary lemmata that we prove next.

First we use Gaussian tail bounds to prove that the largest entry of a Gaussian vector can be quite large with inverse polynomial probability:

Lemma A.16. *Let $x \sim \mathcal{N}(0, \sigma)$ be a Gaussian random variable with mean zero and variance σ^2. Then, for any $t > 0$, we have*

$$\left(\frac{\sigma}{t} - \frac{\sigma^3}{t^3}\right) f(t/\sigma) \leq \Pr[x \geq t] \leq \frac{\sigma}{t} f(t/\sigma),$$

where $f(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$.

Proof: Let $z = \frac{x}{\sigma}$, where $z \sim \mathcal{N}(0,1)$ is a standard Gaussian random variable. Then, we have $\Pr[x \geq t] = \Pr[z \geq t/\sigma]$, and therefore, the result is proved by using standard tail bounds for Gaussian random variable. □

Lemma A.17. *Consider $r = [r_1, r_2, \ldots, r_k]^\top \in \mathbb{R}^k$ as a k-dimensional random Gaussian vector with zero mean and covariance Σ, i.e., $r \sim \mathcal{N}(0, \Sigma)$. For any $k \geq 2$, we have*

$$\Pr\left[r_{(1)} \geq 4\sigma_{\max}\sqrt{\log k}\right] \leq k^{-7}.$$

Proof: From Lemma A.16, for any $i \in [k]$, we have

$$\Pr\left[|r_i| \geq 4\sigma_{\max}\sqrt{\log k}\right] \leq \frac{1}{2\sqrt{2\pi \log k}} k^{-8} \leq k^{-8},$$

where the last inequality is concluded from the fact that $k \geq 2$. The result is then proved by taking a union bound. □

Next we prove a basic fact about spectral norm that is used in the proof of Lemma A.15.

Lemma A.18. *Given $h \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$, let $H = [h|h|\cdots|h]\,\mathrm{Diag}(v) \in \mathbb{R}^{m\times n}$. Then, $\|H\| = \|h\|\|v\|$.*

Proof: By definition

$$\|H\| = \sup_{\|x\|=1} \|Hx\|.$$

We have $Hx = \langle v, x\rangle h$, and therefore, $\|Hx\| = |\langle v, x\rangle|\|h\|$. This is maximized by $x = v/\|v\|$, and this finishes the proof. □

Finally, we show that noise matrix $\Psi(I, I, \theta)$ has bounded norm with high probability which is useful for initialization argument in Theorem A.1.

Lemma A.19. *Let $\theta \in \mathbb{R}^d$ be standard multivariate Gaussian as $\mathcal{N}(0, I_d)$. Then, for any $\alpha_0 > 1$, we have*

$$\Pr\left[\|\Psi(I, I, \theta)\| \leq \alpha_0\sqrt{d}\psi\right] \geq 1 - e^{-(\alpha_0-1)^2 d/2},$$

where $\psi := \|\Psi\|$ is the spectral norm of error tensor Ψ.

Proof: Let $\theta_n := \frac{1}{\|\theta\|}\theta$ denote the normalized version of θ. Then, we have

$$\|\Psi(I, I, \theta)\| = \|\theta\| \cdot \|\Psi(I, I, \theta_n)\| \leq \|\theta\|\psi,$$

where the last inequality is from the definition of tensor spectral norm. Applying the bound on $\|\theta\|$ in Lemma A.20 finishes the proof. □

The following lemma provides concentration bound for the norm of standard Gaussian vector which is basically a tail bound for the chi-squared random variable.

Lemma A.20 (Lemma 15 of Dasgupta et al. [66])**.** *Let the random vector θ is distributed as $\mathcal{N}(0, I_d)$. Then, for any $\alpha_0 > 1$, we have*

$$\Pr\left[\|\theta\| \geq \alpha_0\sqrt{d}\right] \leq e^{-(\alpha_0-1)^2 d/2}.$$

A.5 Clustering Process

In the last step of main algorithm, we need to cluster the generated 4-tuples into k clusters. Theoretically, we only have convergence guarantees when the initialization vectors are good enough, while the other initializations can potentially generate arbitrary 4-tuples. In the worst case, these arbitrary 4-tuples can make the clustering process hard, and therefore, we provide specific Procedure 3 for which the output properties are provided in Lemma A.23.

Note that the key observation for the algorithm is if $T(\widehat{a}, \widehat{b}, \widehat{c})$ is large for some $(\widehat{a}, \widehat{b}, \widehat{c})$, then these vectors are close to (a_i, b_i, c_i) for some $i \in [k]$.

For simplicity, we only prove this when the initialization procedure in Theorem 2.5 takes polynomial time, namely $k = O(d)$ and $w_{\max}/w_{\min} = O(1)$. Without loss of generality, we also assume $w_{\max} = w_1 \geq w_2 \geq \cdots \geq w_k = w_{\min}$. In this case, we choose the threshold ϵ in the following lemmata to be some small constant depending on k/d and $w_{\max}/w_{\min}$. Also, we work in the case when noise $\Psi = 0$, however the proof still works when the noise $\psi = \|\Psi\| = o(1)$.

Lemma A.21. *Suppose*

$$\max\{|\langle a_i, \widehat{a}\rangle|, |\langle b_i, \widehat{b}\rangle|, |\langle c_i, \widehat{c}\rangle|\} \leq \epsilon, \quad \forall i \in [t-1],$$

for some $t \in [k]$. *Let* $\delta := O\left(\frac{w_{\max}}{w_{\min}}\epsilon^{3-p}\right)$, *and assume* $|T(\widehat{a},\widehat{b},\widehat{c})| \geq (1-\delta)w_t$. *Then, there exists some* j *such that*

$$\max\{\mathrm{dist}(\widehat{a},a_j),\mathrm{dist}(\widehat{b},b_j),\mathrm{dist}(\widehat{c},c_j)\} < \frac{w_{\min}}{10w_{\max}}.$$

Proof: Partition tensor $T = \sum_{i\in[k]} w_i a_i \otimes b_i \otimes c_i$ to $T_1 + T_2$, where T_1 contains all the terms indexed from 1 to $t-1$, and T_2 contains the remaining terms. From Corollary A.1, we have

$$|T_1(\widehat{a},\widehat{b},\widehat{c})| \leq w_{\max}\left\|A_{[t-1]}^\top\widehat{a}\right\|_3 \cdot \left\|B_{[t-1]}^\top\widehat{b}\right\|_3 \cdot \left\|C_{[t-1]}^\top\widehat{c}\right\|_3,$$

where $A_{[t-1]} \in \mathbb{R}^{d\times(t-1)}$ denotes the first $t-1$ columns of A, and similarly for $B_{[t-1]}$ and $C_{[t-1]}$. We also have

$$\left\|A_{[t-1]}^\top\widehat{a}\right\|_3^3 \leq \left\|A_{[t-1]}^\top\widehat{a}\right\|_p^p \cdot \max_{i\in[t-1]}|\langle a_i,\widehat{a}\rangle|^{3-p} = O\left(\epsilon^{3-p}\right),$$

where Assumption (A10) and the assumption in the lemma are exploited in the last step. Similar arguments hold for b and c. Combining with the earliest inequality, we have

$$|T_1(\widehat{a},\widehat{b},\widehat{c})| \leq w_{\max}O\left(\epsilon^{3-p}\right) \leq w_t\delta,$$

where the definition of δ is exploited in the last inequality. Applying assumption $|T(\widehat{a},\widehat{b},\widehat{c})| \geq (1-\delta)w_t$ to the above bound, we have

$$|T_2(\widehat{a},\widehat{b},\widehat{c})| \geq (1-2\delta)w_t. \tag{A.19}$$

On the other hand, from Corollary A.1,

$$|T_2(\widehat{a},\widehat{b},\widehat{c})| \leq w_t\|A^\top\widehat{a}\|_3\|B^\top\widehat{b}\|_3\|C^\top\widehat{c}\|_3.$$

Since all the 3-norms are bounded by $1 + o(1)$, each of them must be at least $1 - O(\delta)$ to let inequality (A.19) hold. Now we have

$$1 - O(\delta) \leq \sum_{j=1}^{k} |\langle a_j, \widehat{a}\rangle|^3 \leq \max\{|\langle a_j, \widehat{a}\rangle|\}^{3-p} \sum_{t=1}^{k} |\langle a_j, \widehat{a}\rangle|^p \leq (1 + o(1)) \max\{|\langle a_j, \widehat{a}\rangle|\}^{3-p},$$

where the last inequality is from Assumption (A10). This implies $\max\{|\langle a_j, \widehat{a}\rangle|\} = 1 - O(\delta)$, which in turn implies there exists a j such that

$$\text{dist}(\widehat{a}, a_j) < w_{\min}/10w_{\max}$$

when ϵ and δ are small enough.

By symmetry we know there is also a j' such that $\text{dist}(\widehat{b}, b_{j'}) < w_{\min}/10w_{\max}$. If $j \neq j'$, then it is easy to check $T_2(\widehat{a}, \widehat{b}, \widehat{c})$ cannot be large. Hence, $j = j'$ and the Lemma is correct. □

On the other hand, we know if there is a good initialization, the largest $T(\widehat{a}, \widehat{b}, \widehat{c})$ must be large.

Lemma A.22. *Suppose there exists a good initialization (see initialization condition* (2.14) *in the local convergence theorem) for some column $t \in [k]$, and*

$$\max\{|\langle a_i, \widehat{a}^{(0)}\rangle|, |\langle b_i, \widehat{b}^{(0)}\rangle|, |\langle c_i, \widehat{c}^{(0)}\rangle|\} \leq \epsilon, \quad \forall i \neq t.$$

Let $\delta := O\left(\frac{w_{\max}}{w_{\min}}\epsilon^{3-p}\right)$. Then the corresponding output of iterations in Algorithm 1 denoted by $(\widehat{a}, \widehat{b}, \widehat{c})$ satisfy

$$|T(\widehat{a}, \widehat{b}, \widehat{c})| > (1 - \delta)w_t.$$

Furthermore, for any $i \neq t$, $\max\{|\langle \widehat{a}, a_i\rangle|, |\langle \widehat{b}, b_i\rangle|, |\langle \widehat{c}, c_i\rangle|\} \leq o(\epsilon)$.

Proof: Similar to the proof of Lemma A.21, partition tensor $T = \sum_{i\in[k]} w_i a_i \otimes b_i \otimes c_i$ to $T_2 = w_t a_t \otimes b_t \otimes c_t$ and $T_1 = T - T_2$. Since the initialization is good, by the local convergence result

in Theorem 2.4, we have

$$\text{dist}(\widehat{a}, a_t) \leq \tilde{O}\left(\frac{w_{\max}}{w_{\min}}\frac{\sqrt{k}}{d}\right) \leq o(\delta),$$

where the incoherence condition and $p > 2$ are exploited in the last step. Therefore, $|T_2(\widehat{a}, \widehat{b}, \widehat{c})| \geq (1 - \delta/2)w_t$.

Similar to Lemma A.21, by using Corollary A.1, we have $|T_1(\widehat{a}, \widehat{b}, \widehat{c})| \leq w_t\delta/2$. Applying these bounds, we have

$$|T(\widehat{a}, \widehat{b}, \widehat{c})| \geq |T_2(\widehat{a}, \widehat{b}, \widehat{c})| - |T_1(\widehat{a}, \widehat{b}, \widehat{c})| \geq (1 - \delta)w_t.$$

The last part of the Lemma is trivial because $\text{dist}(\widehat{a}, a_t)$ is small and $\langle a_i, a_t\rangle$ is small by incoherence.

□

Finally we prove the clustering process succeeds.

Lemma A.23. *Procedure 3 outputs k cluster centers that are $\tilde{O}\left(\frac{w_{\max}}{w_{\min}}\frac{\sqrt{k}}{d}\right)$ close to the true components of the tensor.*

Proof: We prove by induction to show that every step of the algorithm correctly computes one component.

Suppose all previously found 4-tuples are $\tilde{O}(w_{\max}\sqrt{k}/w_{\min}d)$ close to some (a_i, b_i, c_i) (notice that this is true at the beginning when no components are found). Let t be the smallest index that has not been found. Then all the remaining 4-tuples satisfy

$$\max\{|\langle a_i, \widehat{a}\rangle|, |\langle b_i, \widehat{b}\rangle|, |\langle c_i, \widehat{c}\rangle|\} \leq \epsilon, \quad \forall i < t.$$

By Lemma A.22 we know there must be a 4-tuple with $|T(\widehat{a}, \widehat{b}, \widehat{c})| > w_t(1-\delta)$. On the other hand, by Lemma A.21 we know the 4-tuple we found must satisfy $\max\{\text{dist}(\widehat{a}, a_j), \text{dist}(\widehat{b}, b_j)\} < w_{\min}/10w_{\max}$ for some j (and this cannot be some j that has already been found). This tuple then satisfies the

conditions of the local convergence Theorem 2.4. Hence, after N iterations it must have converged to (a_j, b_j, c_j). At this step the algorithm successfully found a new component of the tensor.

□

Appendix B

Proofs for Overcomplete CP Tensor Decomposition: Random Components

Proof of Lemma 2.6: Recall that we have updates of the form

$$\tilde{x}^{(t+1)} = A(y^{(t)})^{*2}, \quad w^{(t)} := (y_{-1}^{(t)})^{*2}, \quad y^{(t)} = A^\top x^{(t)}.$$

Let

$$X^{[t]\setminus 1} := \left[x^{(2)}|\dots|x^{(t)}\right],$$

and let the rows of $Y^{[t]}$ are partitioned as the first and the rest of rows as

$$Y^{[t]} = \left[Y_1^{[t]\top} \middle| Y_{-1}^{[t]\top}\right]^\top.$$

Table B.1: Table of parameters and variables. Superscript (t) denotes the variable at t-th iteration.

Variable	Space	Description	Recursion formula
A	$\mathbb{R}^{d\times k}$	mapping matrix in update formula (2.20)	n.a.
$x^{(t)}$	$\mathbb{R}^d$	update variable in (2.20)	$x^{(t+1)} := \frac{A(y^{(t)})^{*2}}{\|A(y^{(t)})^{*2}\|}$
$y^{(t)}$	$\mathbb{R}^k$	intermediate variable in update formula (2.20)	$y^{(t)} := A^\top x^{(t)}$
$\tilde{x}^{(t)}$	$\mathbb{R}^d$	unnormalized version of $x^{(t)}$	$\tilde{x}^{(t+1)} := A(y^{(t)})^{*2}$
$\widehat{x}^{(t)}$	$\mathbb{R}^d$	noisy version of $x^{(t)}$	$\hat{x}^{(t)} = x^{(t)} + \xi^{(t)}$; see (2.27)
$\xi^{(t)}$	$\mathbb{R}^d$	Contribution of noise in tensor power update given noisy tensor $\hat{T} = T + E$	$\hat{x}^{(t)} = x^{(t)} + \xi^{(t)}$; see (2.27)
B	$\mathbb{R}^{d\times(k-1)}$	matrix $A := [a_1\ a_2\ \cdots\ a_k]$ with first column removed, i.e., $B := [a_2\ a_3\ \cdots\ a_k]$. Note that the first column a_1 is the desired one to recover.	n.a.
$B^{(t,1)}$	$\mathbb{R}^{d\times(k-1)}$	conditional distribution of B given previous iterations at the middle of t^{th} iteration (before update step $\tilde{x}^{(t+1)} = A(y^{(t)})^{*2}$)	$B^{(t,1)} \overset{(\mathrm{d})}{=} B\|\{X^{[t]}, Y^{[t]}\}$
$B^{(t,2)}$	$\mathbb{R}^{d\times(k-1)}$	conditional distribution of B given previous iterations at the end of t^{th} iteration (after update step $\tilde{x}^{(t+1)} = A(y^{(t)})^{*2}$)	$B^{(t,2)} \overset{(\mathrm{d})}{=} B\|\{X^{[t+1]}, Y^{[t]}\}$
$B_{\text{res.}}^{(t,1)}$	$\mathbb{R}^{d\times(k-1)}$	residual independent randomness left in $B^{(t,1)}$; see Lemma 2.6.	see equation (2.23)
$B_{\text{res.}}^{(t,2)}$	$\mathbb{R}^{d\times(k-1)}$	residual independent randomness left in $B^{(t,2)}$; see Lemma 2.6.	see equation (2.25)
$w^{(t)}$	$\mathbb{R}^{k-1}$	intermediate variable in update formula (2.20)	$w^{(t)} := (y_{-1}^{(t)})^{*2}$
$u^{(t)}$	$\mathbb{R}^d$	part of $x^{(t)}$ representing the left independent randomness	$u^{(t+1)} := B_{\text{res.}}^{(t,1)} w^{(t)}$
$v^{(t)}$	$\mathbb{R}^{k-1}$	part of $y_{-1}^{(t)}$ representing the left independent randomness	$v^{(t)} := (B_{\text{res.}}^{(t-1,2)})^\top x^{(t)}$

We now make the following simple observations

$$\begin{aligned} B^{(t,1)} &\overset{(\mathrm{d})}{=} B|\{Y^{[t]} = A^\top X^{[t]},\, \tilde{X}^{[t]\setminus 1} = A(Y^{[t-1]})^{*2}\} \\ &\overset{(\mathrm{d})}{=} B|\{Y_{-1}^{[t]} = B^\top X^{[t]},\, \tilde{X}^{[t]\setminus 1} = a_1(Y_1^{[t-1]})^{*2} + BW^{[t-1]}\} \\ &\overset{(\mathrm{d})}{=} B|\{v^{(1)} = B^\top x^{(1)}, \ldots, v^{(t)} = (B_{\text{res.}}^{(t-1,2)})^\top x^{(t)}, \\ &\qquad u^{(2)} = B_{\text{res.}}^{(1,1)} w^{(1)}, \ldots, u^{(t)} = B_{\text{res.}}^{(t-1,1)} w^{(t-1)}\}, \end{aligned}$$

where the second equivalence comes from the fact that B is matrix A with first column removed. Now applying Corollary 2.3, we have the result. The distribution of $B^{(t,2)}$ follow similarly. □

B.1 Analysis of Induction Argument

In this section, we analyze the basis of induction and inductive step for the induction argument proposed in Section 2.6.1.3 for the proof of Lemma 2.3.

B.1.1 Basis of induction

We first show that the hypothesis holds for initialization vector $x^{(1)}$ as the basis of induction.

Claim 5 (Basis of induction)**.** *The induction hypothesis is true for* $t = 1$*.*

Proof: Notice that induction hypothesis for $t = 1$ only involves the bounds on $\|x^{(1)}\|$ and $\langle a_1, x^{(1)}\rangle$ as in Hypotheses 1 and 3, respectively. These bounds are directly argued by the correlation assumption on the initial vector $x^{(1)}$ stated in (2.19) where $\delta_1 = \delta_1^* = \Delta_1^* = 1$. □

B.1.2 Inductive step

Assuming the induction hypothesis holds for all the values till the end of iteration $t-1$ (stated in Section 2.6.1.3), we analyze the t-th iteration of the algorithm, and prove that induction hypothesis also holds for the values at the end of iteration t. See Figure 2.2 where the scope of iteration t and

the flow of the algorithm is shown. In the rest of this section, we pursue the flow of the algorithm at iteration t starting from computing $y^{(t)}$ and ending up with computing $x^{(t+1)}$ to prove the desired induction hypothesis at iteration t.

Hypothesis 4

We start by showing that the induction Hypothesis 4 holds at iteration t using the induction Hypotheses 1 and 2 in the previous iteration.

Claim 6. *We have*

$$\frac{\delta_t}{2}\sqrt{\frac{k}{d}} \leq \|v^{(t)}\| \leq 2\sqrt{\frac{k}{d}},$$
$$\frac{\delta'_t}{2}\frac{\sqrt{k}}{d} \leq \|u^{(t+1)}\| \leq 2\Delta'_t\frac{\sqrt{k}}{d}.$$

Proof: Recall that $v^{(t)} := (B_{\text{res.}}^{(t-1,2)})^\top x^{(t)}$, and by applying the form of $B_{\text{res.}}^{(t-1,2)}$ in (2.25), we have

$$v^{(t)} \overset{\text{(d)}}{=} P_{\perp_{W^{[t-1]}}} B'^\top P_{\perp_{X^{[t-1]}}} x^{(t)}. \tag{B.1}$$

Since random matrix $B' \in \mathbb{R}^{d\times(k-1)}$ is an independent copy of B with i.i.d. Gaussian entries $B'_{ij} \sim \mathcal{N}(0, \frac{1}{d})$, we know $v^{(t)}$ is a random Gaussian vector in the subspace orthogonal to $W^{[t-1]}$. On the other hand, for any vector $z \in \mathbb{R}^d$, we have

$$\mathbb{E}\left[\|B'^\top z\|^2\right] = z^\top \mathbb{E}\left[B'B'^\top\right] z = \frac{k-1}{d}\|z\|^2,$$

where $\mathbb{E}\left[B'B'^\top\right] = \frac{k-1}{d}I$ is exploited. Let $z = P_{\perp_{X^{[t-1]}}} x^{(t)}$. Then, by applying the above equality to the expansion of $v^{(t)}$ in (B.1), we have

$$\mathbb{E}\left[\|v^{(t)}\|^2\right] = \frac{k-t}{k-1}\cdot\frac{k-1}{d}\cdot\|P_{\perp_{X^{[t-1]}}} x^{(t)}\|^2 = \frac{k-t}{d}\cdot\|P_{\perp_{X^{[t-1]}}} x^{(t)}\|^2 \in \left[\delta_t^2\frac{k}{d}\left(1-\frac{t}{k}\right), \frac{k}{d}\right],$$

where $\dim(W^{[t-1]}) = t-1$ is also used in the first step, and the last step is concluded from Hypothesis 1. Finally, by concentration property of random Gaussian vectors, when $t \ll d$ we have with high probability

$$\|v^{(t)}\| \in \left[\frac{\delta_t}{2}\sqrt{\frac{k}{d}}, 2\sqrt{\frac{k}{d}}\right].$$

Similarly, for $u^{(t+1)} := B^{(t,1)}_{\text{res.}} w^{(t)}$, and by applying the form of $B^{(t,1)}_{\text{res.}}$ in (2.23), we have

$$u^{(t+1)} \overset{\text{(d)}}{=} P_{\perp_{X^{[t]}}} \tilde{B} P_{\perp_{W^{[t-1]}}} w^{(t)}. \tag{B.2}$$

Since random matrix $\tilde{B} \in \mathbb{R}^{d\times(k-1)}$ is an independent copy of B with i.i.d. Gaussian entries $\tilde{B}_{ij} \sim \mathcal{N}(0, \frac{1}{d})$, we know $u^{(t+1)}$ is a random Gaussian vector in the subspace orthogonal to $X^{[t]}$. On the other hand, for any vector $z \in \mathbb{R}^{k-1}$, we have

$$\mathbb{E}\left[\|\tilde{B}z\|^2\right] = z^\top \mathbb{E}\left[\tilde{B}^\top \tilde{B}\right] z = \|z\|^2,$$

where $\mathbb{E}\left[\tilde{B}^\top \tilde{B}\right] = I$ is exploited. Let $z = P_{\perp_{W^{[t-1]}}} w^{(t)}$. Then, by applying the above equality to the expansion of $u^{(t+1)}$ in (B.2), we have

$$\mathbb{E}\left[\|u^{(t+1)}\|^2\right] = \frac{d-t}{d} \cdot \|P_{\perp_{W^{[t-1]}}} w^{(t)}\|^2 \in \left[(\delta'_t)^2 \frac{k}{d^2}\left(1 - \frac{t}{d}\right), (\Delta'_t)^2 \frac{k}{d^2}\right],$$

where $\dim(X^{[t]}) = t$ is also used in the first step, and the last step is concluded from Hypothesis 2. Finally, by concentration property of random Gaussian vectors, when $t \ll d$ we have with high probability

$$\|u^{(t+1)}\| \in \left[\frac{\delta'_t}{2}\frac{\sqrt{k}}{d}, 2\Delta'_t \frac{\sqrt{k}}{d}\right].$$

□

Hypothesis 2

Computing $y^{(t)}$: In the first step of iteration t, the algorithm computes $y^{(t)}$. By induction Hypothesis 3, we know $|y_1^{(t)}| = \tilde{\Theta}(d^{\beta 2^{t-1}}\sqrt{k}/d)$. The other coordinates of $y^{(t)} := A^\top x^{(t)}$ are $y_{-1}^{(t)} = B^\top x^{(t)}$ which conditioning on the previous iterations are equivalent (in distribution) to

$$\begin{aligned} y_{-1}^{(t)} &\overset{(\mathrm{d})}{=} \left(B^{(t-1,2)}\right)^\top x^{(t)} \\ &= \left(\sum_{i\in[t-1]} \left(\frac{u^{(i+1)}(P_{\perp_{W^{[i-1]}}} w^{(i)})^\top}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|^2} + \frac{P_{\perp_{X^{[i-1]}}} x^{(i)}(v^{(i)})^\top}{\|P_{\perp_{X^{[i-1]}}} x^{(i)}\|^2}\right) + B_{\text{res.}}^{(t-1,2)}\right)^\top x^{(t)} \\ &= \sum_{i\in[t-1]} \left(\tilde{\Theta}\left(\frac{d^2}{k}\right) P_{\perp_{W^{[i-1]}}} w^{(i)} \langle u^{(i+1)}, x^{(t)}\rangle + \tilde{\Theta}(1) v^{(i)} \langle P_{\perp_{X^{[i-1]}}} x^{(i)}, x^{(t)}\rangle\right) + v^{(t)}, \end{aligned} \tag{B.3}$$

where form of $B^{(t-1,2)}$ in (2.24) is used in the second equality. The bounds on the norms come from Hypotheses 1 and 2. The last term is by definition $v^{(t)} := (B_{\text{res.}}^{(t-1,2)})^\top x^{(t)}$. Note that differences in polylog factors in the (upper and lower) bounds in Hypotheses 1 and 2 are represented by notation $\tilde{\Theta}(\cdot)$.

We will establish subsequently that if $k > d$, the terms involving $v^{(i)}$'s in the above expansion dominate, and the terms involving $P_{\perp_{W^{[i-1]}}} w^{(i)}$'s have norm of a smaller order; see Claim 7.

Computing $w^{(t)}$: In the next step of the algorithm at iteration t, $w^{(t)}$ is computed for which we now argue if the induction hypothesis holds up to iteration t, both lower and upper bounds at iteration t as $\|P_{\perp_{W^{[t-1]}}} w^{(t)}\| \in [\delta_t', \Delta_t']\frac{\sqrt{k}}{d}$ (see induction Hypothesis 2) also hold.

Lower bound: For the lower bound, intuitively the *fresh* random vector $v^{(t)}$ should bring enough randomness into $w^{(t)}$. We formulate that in the following lemma.

Lemma B.1. *Suppose R and R' are two subspaces in $\mathbb{R}^k$ with dimension at most $t \le \frac{k}{16(\log k)^2}$. Let $p \in \mathbb{R}^k$ be an arbitrary vector, $z \in \mathbb{R}^k$ be a uniformly random Gaussian vector in the space orthogonal to R, and finally $w = (p+z) * (p+z)$. Then with high probability, we have*

$$\|P_{\perp_{R'}} w\| \ge \frac{\mathbb{E}[\|z\|^2]}{40\sqrt{k}}.$$

Recall that $w^{(t)} := y_{-1}^{(t)} * y_{-1}^{(t)}$, and $y_{-1}^{(t)}$ is expanded in (B.3) as sum of an arbitrary vector and a random Gaussian vector. Applying above lemma with $R = R' = \text{span}(W^{[t-1]})$, we have with high probability

$$\|P_{\perp_{W^{[t-1]}}} w^{(t)}\| \geq \frac{\mathbb{E}[\|v^{(t)}\|^2]}{40\sqrt{k}} \geq \frac{\delta_t^2}{160}\sqrt{k}/d,$$

where Hypothesis 4 gives lower bound $\|v^{(t)}\| \geq \delta_t/2\sqrt{k/d}$ (used in the second inequality). By choosing $\delta_t' = \delta_t^2/160$ the lower bound in Hypothesis 2 is proved.

Upper bound: In order to prove the upper bounds in Hypothesis 2, we follow the sequence of arguments below:

$$\boxed{\text{Claim 7: } \|y_{-1}^{(t)}\|_\infty \overset{(\cdot)^2}{\Longrightarrow} \|w^{(t)}\|_\infty \overset{\text{Lemma B.2}}{\Longrightarrow} \|P_{\perp_{W^{[t-1]}}} w^{(t)}\|_\infty \Rightarrow \|P_{\perp_{W^{[t-1]}}} w^{(t)}\|}$$

First we prove a bound on the infinity norm of $y_{-1}^{(t)}$:

Claim 7 (Upper bound on $\|y_{-1}^{(t)}\|_\infty$)**.** *We have*

$$\|y_{-1}^{(t)}\|_\infty \leq \frac{t}{\delta_t}\frac{\log d}{\sqrt{d}} + (t-1)\left(\frac{\Delta_{t-1}'}{\delta_{t-1}'}\right)^2 \frac{1}{\sqrt{k}} = \tilde{O}\left(\frac{1}{\sqrt{d}}\right).$$

Proof: We exploit the induction hypothesis to bound the ℓ_∞ norm of all the terms in the expansion of $y_{-1}^{(t)}$ in (B.3).

For the terms involving $v^{(i)}$, since they are random Gaussian vectors with expected square norm at most k/d, by Lemma B.4 we know $\|v^{(i)}\|_\infty \leq \frac{\log d}{\sqrt{d}}$ with high probability. In addition, for $v^{(i)}$, $i < t$, the coefficient is bounded as

$$\frac{\langle P_{\perp_{X^{[i-1]}}} x^{(i)}, x^{(t)}\rangle}{\|P_{\perp_{X^{[i-1]}}} x^{(i)}\|^2} \leq \frac{1}{\|P_{\perp_{X^{[i-1]}}} x^{(i)}\|} \leq \frac{1}{\delta_i}, \tag{B.4}$$

where the last step uses Hypothesis 1. Therefore, the total contribution from terms involving $v^{(i)}$ in $\|y_{-1}^{(t)}\|_\infty$ is bounded by $\frac{t}{\delta_t}\frac{\log d}{\sqrt{d}}$.

For the terms involving $P_{\perp_{W^{[i-1]}}} w^{(i)}, i \in [t-1]$, we have from Hypothesis 2 that the ℓ_∞ norm is bounded as $\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|_\infty \leq \Delta'_i \frac{1}{d}$. In addition, the corresponding coefficient is bounded by

$$\frac{\langle u^{(i+1)}, x^{(t)} \rangle}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|^2} \leq \frac{\|u^{(i+1)}\| \cdot \|x^{(t)}\|}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|^2} \leq \frac{2\Delta'_i}{{\delta'_i}^2} \frac{d}{\sqrt{k}}. \tag{B.5}$$

Again bounds in Hypotheses 2 and 4 are exploited in the last inequality. Hence, the total contribution from terms involving $P_{\perp_{W^{[i-1]}}} w^{(i)}, i \in [t-1]$ in $\|y_{-1}^{(t)}\|_\infty$ is bounded by $(t-1)\left(\frac{\Delta'_{t-1}}{\delta'_{t-1}}\right)^2 \frac{1}{\sqrt{k}}$.

Combining the above bounds finishes the proof. □

Since $w^{(t)} := y_{-1}^{(t)} * y_{-1}^{(t)}$, the above claim immediately implies that

$$\|w^{(t)}\|_\infty \leq \tilde{O}\left(\frac{1}{d}\right). \tag{B.6}$$

Now we have the ℓ_∞ norm on w, however we need to bound the ℓ_∞ norm of the projected vector $P_{\perp_{W^{[t-1]}}} w^{(t)}$. Intuitively this is clear as the vectors in the space $W^{[t-1]}$ all have small ℓ_∞ as guaranteed by induction hypothesis. We formalize this intuition using the following lemma.

Lemma B.2. *Suppose R is a subspace in $\mathbb{R}^k$ of dimension t', such that there is a basis $\{r_1, \ldots, r_{t'}\}$ with $\|r_i\|_\infty \leq \frac{\Delta}{\sqrt{k}}$ and $\|r_i\| = 1$. Let $p \in \mathbb{R}^k$ be an arbitrary vector, then*

$$\|P_{\perp_R} p\|_\infty \leq \|p\|_\infty + \|p\| \Delta \frac{\sqrt{t'}}{\sqrt{k}}.$$

Let $R = \text{span}(W^{[t-1]})$. Then the vectors $P_{\perp_{W^{[i-1]}}} w^{(i)} / \|P_{\perp_{W^{[i-1]}}} w^{(i)}\|$, $i \in [t-1]$ form a basis for subspace R, and we know from Hypothesis 2 that the ℓ_∞ norm of each of these basis vectors is bounded by $\frac{\Delta}{\sqrt{k}}$ for $\Delta := \frac{\Delta'_{t-1}}{\delta'_{t-1}}$ which is of order $\text{polylog}\, d$. Applying above lemma, we have

$$\|P_{\perp_{W^{[t-1]}}} w^{(t)}\|_\infty \leq \|w^{(t)}\|_\infty (1 + \Delta\sqrt{t-1}) \leq \frac{\Delta'_t}{d},$$

where the last inequality uses bound (B.6), and appropriate choosing for Δ'_t which is of order $\text{polylog}\, d$ and only depends on t and $\log d$. This concludes the upper bound on the ℓ_∞ norm in

Hypothesis 2. The upper bound on the ℓ_2 norm is also immediately argued using this ℓ_∞ norm bound where an additional $\sqrt{k}$ factor shows up.

Hypothesis 1

Computing $x^{(t+1)}$:

In the next step of iteration t, the algorithm computes $x^{(t+1)}$. Conditioning on the previous iterations, the unnormalized version $\tilde{x}^{(t+1)}$ is equivalent (in distribution) to

$$\begin{aligned}\tilde{x}^{(t+1)} &\overset{\text{(d)}}{=} B^{(t,1)}w^{(t)} + (y_1^{(t)})^2 a_1 \\ &= \sum_{i\in[t-1]} \frac{u^{(i+1)}(P_{\perp_{W^{[i-1]}}} w^{(i)})^\top}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|^2} w^{(t)} + \sum_{i\in[t]} \frac{P_{\perp_{X^{[i-1]}}} x^{(i)}(v^{(i)})^\top}{\|P_{\perp_{X^{[i-1]}}} x^{(i)}\|^2} w^{(t)} + B^{(t,1)}_{\text{res.}} w^{(t)} + (y_1^{(t)})^2 a_1 \\ &= \sum_{i\in[t-1]} \tilde{\Theta}\left(\frac{d^2}{k}\right) u^{(i+1)} \langle P_{\perp_{W^{[i-1]}}} w^{(i)}, w^{(t)}\rangle + \sum_{i\in[t]} \tilde{\Theta}(1) P_{\perp_{X^{[i-1]}}} x^{(i)} \langle v^{(i)}, w^{(t)}\rangle + u^{(t+1)} + (y_1^{(t)})^2 a_1,\end{aligned} \tag{B.7}$$

where form of $B^{(t,1)}$ in (2.22) is used in the second equality. The bounds on the norms come from Hypotheses 1 and 2. The last term is the definition of $u^{(t+1)} := B^{(t,1)}_{\text{res.}} w^{(t)}$. Note that differences in polylog factors in the (upper and lower) bounds in Hypotheses 1 and 2 are represented by notation $\tilde{\Theta}(\cdot)$.

The goal is to prove Hypothesis 1 holds at t-th iteration (which is to show the desired lower and upper bounds on $\|P_{\perp_{X^{[t]}}} x^{(t+1)}\|$) assuming induction hypothesis holds for earlier iterations. Given the normalization $x^{(t+1)} := \tilde{x}^{(t+1)}/\|\tilde{x}^{(t+1)}\|$ in each iteration, we have

$$\|P_{\perp_{X^{[t]}}} x^{(t+1)}\| = \frac{1}{\|\tilde{x}^{(t+1)}\|} \|P_{\perp_{X^{[t]}}} \tilde{x}^{(t+1)}\|. \tag{B.8}$$

Therefore, we first bound the norm of $\tilde{x}^{(t+1)}$ which turns out to be $\|\tilde{x}^{(t+1)}\| = \tilde{\Theta}\left(\frac{\sqrt{k}}{d}\right)$ as argued in the following.

Lower bound: The lower bound on $\|\tilde{x}^{(t+1)}\|$ simply follows from the term $u^{(t+1)}$, which is an independent random Gaussian.

Claim 8. *If $t \leq \frac{d}{10}$, then we have whp*

$$\|\tilde{x}^{(t+1)}\| \geq \frac{\delta'_t}{4}\frac{\sqrt{k}}{d}.$$

Proof: We have

$$\|\tilde{x}^{(t+1)}\| \geq \|P_{\text{span}(X^{[t]},U^{[t]},a_1)^\perp}\tilde{x}^{(t+1)}\| = \|P_{\text{span}(X^{[t]},U^{[t]},a_1)^\perp}u^{(t+1)}\|.$$

Note that the equality is concluded from expansion of $\tilde{x}^{(t+1)}$ in (B.7) where all the components of $\tilde{x}^{(t+1)}$ in the subspace $\text{span}(X^{[t]}, U^{[t]}, a_1)^\perp$ is represented by $u^{(t+1)}$. The vector $P_{\text{span}(X^{[t]},U^{[t]},a_1)^\perp}u^{(t+1)}$ is the projection of a random Gaussian vector $u^{(t+1)}$ in to a subspace of dimention $d - o(d)$. Hence it is still a random Gaussian vector with expected square norm larger than $\frac{\delta'^2_t}{2}\frac{k}{d^2}$. By Lemma B.3, with high probability the desired bound holds. □

Upper bound: The upper bound is argued in the following claim.

Claim 9. *We have either*

$$\langle x^{(t+1)}, a_1\rangle \geq 1 - \gamma,$$

for some constant $\gamma > 0$ or

$$\|\tilde{x}^{(t+1)}\| \leq \tilde{O}\left(\frac{\sqrt{k}}{d}\right).$$

Proof: Let $\tilde{x}^{(t+1)}$ in (B.7) be written as $\tilde{x}^{(t+1)} = z + (y_1^{(t)})^2 a_1$ where vector $z \in \mathbb{R}^d$ represents all the other terms in the expansion. The analysis is done under two cases 1) $(y_1^{(t)})^2 \geq \frac{2}{\gamma}\|z\|$ and 2) $(y_1^{(t)})^2 < \frac{2}{\gamma}\|z\|$ for some constant $\gamma > 0$. Note that the left hand side is the norm of $(y_1^{(t)})^2 a_1$ since $\|a_1\| = 1$, and in addition $(y_1^{(t)})^2 = \langle x^{(t)}, a_1\rangle^2$.

Case 1 $\left((y_1^{(t)})^2 \geq \frac{2}{\gamma}\|z\|\right)$: For the $x^{(t+1)} := \tilde{x}^{(t+1)}/\|\tilde{x}^{(t+1)}\|$, we have

$$\begin{aligned}\langle x^{(t+1)}, a_1\rangle &= \frac{1}{\|z + (y_1^{(t)})^2 a_1\|}\langle z + (y_1^{(t)})^2 a_1, a_1\rangle \\ &\geq \frac{1}{\|z\| + (y_1^{(t)})^2}\left[(y_1^{(t)})^2 - \|z\|\right] \\ &\geq \frac{1 - \frac{\gamma}{2}}{1 + \frac{\gamma}{2}} \geq 1 - \gamma,\end{aligned}$$

where triangle and Cauchy-Schwartz inequality are used in the first bound, and the second inequality is concluded from assumption $(y_1^{(t)})^2 \geq \frac{2}{\gamma}\|z\|$.

Case 2 $\left((y_1^{(t)})^2 < \frac{2}{\gamma}\|z\|\right)$: We exploit the induction hypothesis to bound the norm of all the terms in the expansion of $\tilde{x}^{(t+1)}$ in (B.7).

For the terms involving $u^{(i+1)}, i \in [t]$, we have $\|u^{(i+1)}\| \leq 2\Delta_i'\frac{\sqrt{k}}{d}$ from Hypothesis 4 and the argument for $\|u^{(t+1)}\|$. In addition, for $u^{(i+1)}$, $i \in [t-1]$, the coefficient is bounded as

$$\frac{\langle P_{\perp_{W^{[i-1]}}} w^{(i)}, w^{(t)}\rangle}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|^2} \leq \frac{\|w^{(t)}\|}{\|P_{\perp_{W^{[i-1]}}} w^{(i)}\|} \leq \frac{\Delta_t'}{\delta_i'}, \tag{B.9}$$

where Cauchy-Schwartz inequality is used in the first bound, and the bound in Hypothesis 2 and (B.6) are exploited in the last inequality. Therefore, the total contribution from terms involving $u^{(i+1)}$ in $\|\tilde{x}^{(t+1)}\|$ is bounded by $\frac{2(t-1)\Delta_t'^2}{\delta_t'}\frac{\sqrt{k}}{d}$.

For the terms involving $P_{\perp_{X^{[i-1]}}} x^{(i)}, i \in [t]$, we have $\|P_{\perp_{X^{[i-1]}}} x^{(i)}\| \leq 1$, but the coefficient $\langle v^{(i)}, w^{(t)}\rangle$ needs further analysis to be bounded which is done in Lemma B.3 saying $|\langle v^{(i)}, w^{(t)}\rangle| \leq \tilde{O}\left(\frac{\sqrt{k}}{d}\right)$. This implies that the total contribution from terms involving $P_{\perp_{X^{[i-1]}}} x^{(i)}$ in $\|\tilde{x}^{(t+1)}\|$ is bounded by $\tilde{O}\left(\frac{\sqrt{k}}{d}\right)$.

Combining the above bounds and considering the assumption that the norm of $(y_1^{(t)})^2 a_1$ in the expansion of $\tilde{x}^{(t+1)}$ is dominated by the norm of other terms argued above, the proof is complete concluding that $\|\tilde{x}^{(t+1)}\| \leq \tilde{O}\left(\frac{\sqrt{k}}{d}\right)$. □

Lemma B.3. *Under the induction hypothesis (up to update step $\tilde{x}^{(t+1)} := A(y^{(t)})^{*2}$ at iteration t), we have for $i \in [t]$,*

$$|\langle v^{(i)}, w^{(t)} \rangle| \leq O\left(t^3 \frac{(\Delta'_{t-1})^4}{(\delta'_{t-1})^4 \delta_t^2} (\log d) \frac{\sqrt{k}}{d} \right) = \tilde{O}\left(\frac{\sqrt{k}}{d} \right).$$

Using (B.8) and the fact that $\|\tilde{x}^{(t+1)}\| = \tilde{\Theta}\left(\frac{\sqrt{k}}{d}\right)$, we have

$$\|P_{\perp_{X^{[t]}}} x^{(t+1)}\| \geq \tilde{\Theta}\left(\frac{d}{\sqrt{k}} \right) \|P_{\mathrm{span}(X^{[t]}, U^{[t]}, a_1)^\perp} u^{(t+1)}\| \geq \frac{\delta'_t}{4},$$

where the bound $\|P_{\mathrm{span}(X^{[t]}, U^{[t]}, a_1)^\perp} u^{(t+1)}\| \geq \frac{\delta'_t}{4} \frac{\sqrt{k}}{d}$ is also used. This finishes the proof that Hypothesis 1 holds.

Hypothesis 3

Finally we prove Hypothesis 3 at iteration t given earlier induction hypothesis. The first part of the hypothesis is proved in the following claim.

Claim 10. *We have*

$$|\langle a_1, x^{(t+1)} \rangle| \in [\delta^*_{t+1}, \Delta^*_{t+1}] d^{\beta 2^t} \frac{\sqrt{k}}{d}.$$

Proof: We first show the correlation bound on the unnormalized version as $\langle a_1, \tilde{x}^{(t+1)} \rangle$. Looking at the expansion of $\tilde{x}^{(t+1)}$ in (B.7), the correlation $\langle a_1, \tilde{x}^{(t+1)} \rangle$ involves three types of terms emerging from $(y_1^{(t)})^2 a_1$, $u^{(i+1)}$ and $P_{\perp_{X^{(i-1)}}} x^{(i)}$. In the following, we argue the correlation from each of these terms where we observe that the correlation is dominated by the term $(y_1^{(t)})^2 a_1$, and the rest of terms contribute much smaller amount.

For the term $(y_1^{(t)})^2 a_1$, we have

$$\langle a_1, (y_1^{(t)})^2 a_1 \rangle = (y_1^{(t)})^2 \in [(\delta^*_t)^2, (\Delta^*_t)^2] d^{\beta 2^t} \frac{k}{d^2},$$

where the last part exploits induction Hypothesis 3 in the previous iteration.

For the terms involving $u^{(i+1)}$, these vectors are random Gaussian vectors in a subspace (with dimension $\Omega(d)$), and therefore, we have with high probability

$$\langle a_1, u^{(i+1)} \rangle \leq \mathbb{E}[\|u^{(i+1)}\|] \cdot O\left(\frac{\log d}{\sqrt{d}}\right) \leq \tilde{O}\left(\frac{\sqrt{k}}{d\sqrt{d}}\right) \leq \tilde{O}\left(\frac{k}{d^2}\right),$$

where the correlation bound between two independent random Gaussian vectors in $\Omega(d)$-dimension is used in the first inequality [1], Hypothesis 4 is exploited in the second inequality, and finally last inequality is from assumption $k > d$. In addition, the coefficient associated with $u^{(i+1)}$ is bounded by Δ_t'/δ_i' argued in (B.9). Hence, the total contribution from terms involving $u^{(i+1)}$ in $\langle \tilde{x}^{(t+1)}, a_1 \rangle$ is bounded by $\tilde{O}\left(\frac{k}{d^2}\right)$.

For the terms involving $P_{\perp_{X^{[i-1]}}} x^{(i)}$, by Hypothesis 3 we have

$$\langle a_1, P_{\perp_{X^{[i-1]}}} x^{(i)} \rangle \leq \Delta_i^* d^{\beta 2^{i-1}} \frac{\sqrt{k}}{d}.$$

In addition, the associated coefficient is bounded by $\tilde{O}\left(\frac{\sqrt{k}}{d}\right)$ from Lemma B.3. Hence, the total contribution from terms involving $P_{\perp_{X^{[i-1]}}} x^{(i)}$ in $\langle \tilde{x}^{(t+1)}, a_1 \rangle$ is bounded by $\tilde{O}\left(d^{\beta 2^{t-1}} \frac{k}{d^2}\right)$.

Combining the above bounds implies

$$|\langle a_1, \tilde{x}^{(t+1)} \rangle| \leq \tilde{O}\left(d^{\beta 2^t} \frac{k}{d^2}\right).$$

Finally, using the bound on the norm of $\tilde{x}^{(t+1)}$ argued as $\|\tilde{x}^{(t+1)}\| = \tilde{\Theta}\left(\frac{\sqrt{k}}{d}\right)$ finishes the proof. □

To prove the last part of Hypothesis 3, we use the following lemma which is very similar to Lemma B.2.

Lemma B.4. *Suppose R is a subspace in $\mathbb{R}^d$ of dimension t', such that there is a basis $\{r_1, \ldots, r_{t'}\}$ with $|\langle r_i, a_1 \rangle| \leq \Delta$ and $\|r_i\| = 1$. Let $p \in \mathbb{R}^d$ be an arbitrary vector, then*

$$|\langle P_{\perp_R} p, a_1 \rangle| \leq |\langle p, a_1 \rangle| + \|p\| \Delta \sqrt{t'}.$$

[1] For two independent random Gaussian vectors $p, q \in \mathbb{R}^d$, we have with high probability $\langle p, q \rangle \leq \mathbb{E}[\|p\|] \cdot \mathbb{E}[\|q\|] \cdot O\left(\frac{\log d}{\sqrt{d}}\right)$.

We apply this lemma with $R = \text{span}(X^{[t]})$, and the basis is $P_{\perp X^{[i-1]}} X^{(i)} / \|P_{\perp X^{[i-1]}} X^{(i)}\|$. By induction hypothesis Δ in the lemma is at most $\Delta_*^t d^{\beta 2^t} \sqrt{k}/d$, let $v = x^{(t+1)}$ then this gives the desired bound.

Let $R = \text{span}(X^{[t]})$. Then the vectors $P_{\perp_{X^{[i-1]}}} x^{(i)} / \|P_{\perp_{X^{[i-1]}}} x^{(i)}\|$, $i \in [t]$ form a basis for subspace R, and we know from Hypotheses 1 and 3 that the correlation between these basis vectors and a_1 is bounded by $\Delta := \Delta_t^* d^{\beta 2^{t-1}} \frac{\sqrt{k}}{d}$. Applying above lemma, we have

$$|\langle P_{\perp_{X^{[t]}}} x^{(t+1)}, a_1 \rangle| \leq |\langle x^{(t+1)}, a_1 \rangle| + \Delta\sqrt{t} \leq \Delta_{t+1}^* d^{\beta 2^t} \frac{\sqrt{k}}{d},$$

where the last inequality uses the first part of Hypothesis 3 proved earlier in this section. Note that Δ_{t+1}^* is a new polylog factor here.

B.1.3 Growth rate of δ_t, δ_t', Δ_t', δ_t^*, Δ_t^*

We know that if the number of iterations t is a constant, then the δ and Δ parameters (i.e., δ_t, δ_t', Δ_t', δ_t^*, Δ_t^*) in the induction hypothesis are bounded by polylog factors of d. Here, we show that these parameters can be still bounded even when the number of steps is slightly larger than a constant. Let

$$R_t := \max\{1/\delta_t, \Delta_{t-1}'/\delta_{t-1}', \Delta_t^*/\delta_t^*\}.$$

We know $R_1 = 1$, and by the inductive step analysis we have the following polynomial recursion property.

Claim 11. $R_{t+1} = \text{poly}(R_t, t, \log d)$.

This claim follows from the proof of inductive step, where in every step the δ and Δ parameters are bounded by polynomial functions of previous δ's (Δ's), t, and $\log d$.

We now solve this recursion as follows.

Lemma B.1. *Suppose* $R_{t+1} \leq c_0 R_t^{c_1} t^{c_2} (\log d)^{c_3}$ *where* c_0, c_1, c_2, c_3 *are positive constants, and we know* $R_1 = 1$. *Then*

$$R_t \leq (\log d)^{2^{c_4 t}},$$

for some constant $c_4 > 0$ *depending on* c_0, c_1, c_2, c_3.

Proof: Without loss of generality assume $c_0 \geq 1$, $c_2 \geq 1$, $c_3 \geq 1$, and $R_1 \geq \log d$. Given these assumptions, we have $R_t \geq \max\{c_0, t, \log d\}$, for $t \geq 1$. Applying this to the assumption $R_{t+1} \leq c_0 R_t^{c_1} t^{c_2} (\log d)^{c_3}$, we have

$$R_{t+1} \leq R_t^{1+c_1+c_2+c_3}. \tag{B.10}$$

Pick some $q > 0$ such that $R_1 \leq (\log d)^{2^q}$, and pick some

$$c_4 \geq \max\{q, \log_2(1 + c_1 + c_2 + c_3)\}.$$

Now we prove the result by the induction argument. Since $c_4 \geq q$, the basis of induction holds for R_1. As the inductive step, suppose $R_t \leq (\log d)^{2^{c_4 t}}$. Applying this to (B.10), we have

$$R_{t+1} \leq (\log d)^{(1+c_1+c_2+c_3)2^{c_4 t}} \leq (\log d)^{2^{c_4(t+1)}},$$

where $2^{c_4} \geq (1 + c_1 + c_2 + c_3)$ is used in the last inequality. This finishes the inductive step and the result is proved. □

Using the above bound, we show in the following corollary that the δ and Δ parameters in the induction hypothesis are bounded by polylog factors of d even if the number of steps t goes up to $c \log\log d$ for small enough constant c. In addition, we show that if $\beta \geq (\log d)^{-c_5}$ for some constant $c_5 > 0$, then the power method converges to a point $x^{(t)}$ which is constant close to the true component.

Corollary B.1. *There exists a universal constant $c_5 > 0$ such that if*

$$\beta \geq (\log d)^{-c_5},$$

and the initial correlation is lower bounded by $d^{\beta}\frac{\sqrt{k}}{d}$ (see (2.19)), then with high probability the power method gets to a point that is constant close to the true component in $\Theta(\log \log d)$ number of steps.

Proof: Pick the number of steps to be $t = (\log \log d)/2c_4$, where c_4 is the constant in Lemma B.1. Then, from Lemma B.1 we have

$$R_t \leq (\log d)^{\sqrt{\log d}} \leq o(d),$$

where the last inequality can be shown by taking the log of both sides. This says that the analysis of inductive step still holds after such number of iterations.

Finally, by progress bound in (2.26), we can see that if $\beta \geq (\log d)^{-c_5}$, then the power method converges to a point $x^{(t)}$ which is constant close to the true component. □

B.2 Auxiliary Lemmas for Induction Argument

In this section we prove the lemmas used in arguing inductive step in Appendix B.1.2.

We first introduce the following lemma proposing a lower bound on the singular value of product of matrices.

Lemma B.2 (Merikoski and Kumar 125)**.** *Let C and D be $k \times k$ matrices. If $1 \leq i \leq k$ and $1 \leq l \leq k - i + 1$, then*

$$\sigma_i(CD) \geq \sigma_{i+l-1}(C) \cdot \sigma_{k-l+1}(D),$$

where $\sigma_j(C)$ denotes the j-th singular value (in decreasing order) of matrix C.

B.2.1 Properties of random Gaussian vectors

We start with some basic properties of random Gaussian vectors. First as a simple fact, the norm of a random Gaussian vector is concentrated as follows which is proved via simple concentration inequalities.

Lemma B.3. *Let $z \in \mathbb{R}^d$ be a random Gaussian vector with $\mathbb{E}[zz^\top] = \frac{1}{d}I$. Then we have with high probability $\frac{1}{2} \leq \|z\| \leq 2$.*

Next we show the ℓ_∞ norm of a Gaussian vector is small, even if it is projected on some subspace.

Lemma B.4. *Let R be any linear subspace in $\mathbb{R}^d$ and $z \in \mathbb{R}^d$ be a random Gaussian vector with $\mathbb{E}[zz^T] = \frac{1}{d}I$. Then we have with high probability $\|P_{\perp_R} z\|_\infty \leq \frac{\log d}{\sqrt{d}}$.*

Proof: Since $P_{\perp_R}$ is a projection matrix, in particular the norm of its columns is bounded by 1. Hence, each entry of $P_{\perp_R} z$ is a Gaussian random variable with variance bounded by $\frac{1}{d}$ implying that with high probability the absolute value of each coordinate is smaller than $\frac{\log d}{\sqrt{d}}$. Finally, the desired ℓ_∞ norm bound is argued by applying union bound. □

We can also show that most of the entries are of size at least $\frac{1}{\sqrt{d}}$.

Lemma B.5. *Let R be any linear subspace in $\mathbb{R}^d$ with dimension $t \leq \frac{d}{16(\log d)^2}$ and $z \in \mathbb{R}^d$ be a random Gaussian vector with $\mathbb{E}[zz^T] = \frac{1}{d}I$. Then we have with high probability at least $1/2$ of the entries $i \in [d]$ satisfy $|(P_{\perp_R} z)_i| \geq \frac{1}{4\sqrt{d}}$.*

Proof: Since the entries of z are independent Gaussian random variables with standard deviation $\frac{1}{\sqrt{d}}$, we know with high probability at least $1/2$ of the entries have absolute value larger than $\frac{1}{2\sqrt{d}}$. On the other hand, $P_R z$ is also a random Gaussian vector with expected square norm bounded by

$$\mathbb{E}[\|P_R z\|^2] \leq \frac{\mathbb{E}[\|z\|^2]}{16(\log d)^2} = \frac{1}{16(\log d)^2},$$

where the assumption on the dimension of subspace R is used in the inequality. By Lemma B.4 we know with high probability entries of $P_R z$ are bounded by $1/4\sqrt{d}$. Now $P_{\perp_R} z = z - P_R z$ must have at least $1/2$ of the entries with absolute value larger than $1/4\sqrt{d}$. □

Using the above lemmas, we can prove Lemma B.1.

Lemma B.1 (Restated)**.** *Suppose R and R' are two subspaces in $\mathbb{R}^k$ with dimension at most $t \le \frac{k}{16(\log k)^2}$. Let $p \in \mathbb{R}^k$ be an arbitrary vector, $z \in \mathbb{R}^k$ be a uniformly random Gaussian vector in the space orthogonal to R, and finally $w = (p+z) * (p+z)$. Then with high probability, we have*

$$\|P_{\perp_{R'}} w\| \ge \frac{\mathbb{E}[\|z\|^2]}{40\sqrt{k}}.$$

Proof: Let z, z' be two independent samples of z, and w, w' be the corresponding w vectors. We have

$$w - w' = (p+z) * (p+z) - (p+z') * (p+z') = (2p + z + z') * (z - z'). \tag{B.11}$$

By properties of Gaussian vectors, $z + z'$, $z - z'$ are two *independent* random Gaussian vectors in the subspace orthogonal to R each with expected square norm $2\mathbb{E}[\|z\|^2]$. We use $z_1 := z + z'$ and $z_2 := z - z'$ to denote these two random Gaussian vectors.

Next, we show that with high probability

$$\|P_{\perp_{R'}}(w - w')\| \ge \frac{\mathbb{E}[\|z\|^2]}{20\sqrt{k}}.$$

Note that this implies the result of lemma as follows. Suppose $\|P_{\perp_{R'}} w\| < \frac{1}{40}\mathbb{E}[\|z\|^2]/\sqrt{k}$ with probability δ. Since w' is an independent sample, with probability δ^2 this bound holds for both w and w'. When this happens, we have $\|P_{\perp_{R'}}(w - w')\| < \frac{1}{20}\mathbb{E}[\|z\|^2]/\sqrt{k}$ by triangle inequality. Since we showed δ^2 is negligible, δ is also negligible.

First we sample z_2. Let $R'' = \mathrm{span}(R', p * z_2)$. Then by expansion of $w - w'$ in (B.11), we have

$$\|P_{\perp_{R'}}(w - w')\| = \|P_{\perp_{R'}}\big(2(p * z_2) + (z_1 * z_2)\big)\| \ge \|P_{\perp_{R''}}(z_1 * z_2)\| = \|P_{\perp_{R''}} \mathrm{Diag}(z_2) P_{\perp_R} z_1\|, \tag{B.12}$$

where the inequality is concluded by ignoring the component along $p * z_2$ direction. The last equality is from [2] $u * v = \mathrm{Diag}(u) \cdot v$ (for two vectors u and v), and the assumption that $z_1 = z + z'$ is in the subspace orthogonal to R. For the matrix $P_{\perp_{R''}} \mathrm{Diag}(z_2) P_{\perp_R}$, we have [3]

$$\sigma_{k/4}\left(P_{\perp_{R''}} \mathrm{Diag}(z_2) P_{\perp_R}\right) \geq \sigma_{k/2}\left(\mathrm{Diag}(z_2)\right) \cdot \sigma_{7k/8}\left(P_{\perp_R}\right) \cdot \sigma_{7k/8}\left(P_{\perp_{R''}}\right) \geq \frac{\sqrt{\mathbb{E}[\|z\|^2]}}{4\sqrt{k}},$$

where the first inequality is from Lemma B.2, and the last step is argued as follows. By Lemma B.3, with high probability z_2 has square norm at least $\mathbb{E}[\|z_2\|^2]/2 = \mathbb{E}[\|z\|^2]$, and therefore, by Lemma B.5 at least $k/2$ of its entries have absolute value larger than $\frac{1}{4}\sqrt{\mathbb{E}[\|z\|^2]}/\sqrt{k}$. Therefore, we can restrict attention to the space spanned by the $k/4$ top singular vectors. In addition, within this subspace we have with high probability $\|z_1\|^2 \geq \mathbb{E}[\|z\|^2]/8$, and hence,

$$\|P_{\perp_{R''}} \mathrm{Diag}(z_2) P_{\perp_R} z_1\| \geq \frac{\mathbb{E}[\|z\|^2]}{20\sqrt{k}},$$

which finishes the proof by applying (B.12). □

B.2.2 Properties of projections

In this part we prove some basic properties of projections. Intuitively, if the whole subspace has small inner-product with some vector, then the projection of an arbitrary vector to the orthogonal subspace should not change the inner-product with that particular vector by too much. This is what we require in Lemma B.4.

Lemma B.4 (Restated)**.** *Suppose R is a subspace in $\mathbb{R}^d$ of dimension t', such that there is a basis $\{r_1, \ldots, r_{t'}\}$ with $|\langle r_i, a_1\rangle| \leq \Delta$ and $\|r_i\| = 1$. Let $p \in \mathbb{R}^d$ be an arbitrary vector, then*

$$|\langle P_{\perp_R} p, a_1\rangle| \leq |\langle p, a_1\rangle| + \|p\|\Delta\sqrt{t'}.$$

[2] For vector u, $\mathrm{Diag}(u)$ denotes the diagonal matrix with u as its main diagonal.
[3] Recall that $\sigma_l(A)$ denotes the l-th singular value (in decreasing order) of matrix A.

Proof: We have $P_{\perp_R} p = p - \sum_{i=1}^{t'} \langle p, r_i \rangle r_i$, and therefore

$$\begin{aligned}
|\langle P_{\perp_R} p, a_1 \rangle| &\leq |\langle p, a_1 \rangle| + \sum_{i=1}^{t'} |\langle p, r_i \rangle \langle a_1, r_i \rangle| \\
&\leq |\langle p, a_1 \rangle| + \Delta \sum_{i=1}^{t'} |\langle p, r_i \rangle| \\
&\leq |\langle p, a_1 \rangle| + \Delta \sqrt{t' \sum_{i=1}^{t'} \langle p, r_i \rangle^2} \\
&\leq |\langle p, a_1 \rangle| + \Delta \|p\| \sqrt{t'}.
\end{aligned}$$

The first step is triangle inequality and the third is Cauchy-Schwartz. □

Lemma B.2 is very similar.

Lemma B.2 (Restated)**.** *Suppose R is a subspace in $\mathbb{R}^k$ of dimension t', such that there is a basis $\{r_1, \ldots, r_{t'}\}$ with $\|r_i\|_\infty \leq \frac{\Delta}{\sqrt{k}}$ and $\|r_i\| = 1$. Let $p \in \mathbb{R}^k$ be an arbitrary vector, then*

$$\|P_{\perp_R} p\|_\infty \leq \|p\|_\infty + \|p\| \Delta \frac{\sqrt{t'}}{\sqrt{k}}.$$

This lemma essentially follows from Lemma B.4, because ℓ_∞ norm is just the maximum inner-product to a basis vector. More specifically, the above lemma is applied for all $a_1 = e_j, j \in [k]$, where e_j denotes the j-th basis vector in $\mathbb{R}^k$.

B.2.3 Bounding correlation between v and w

We are only left with Lemma B.3. The main difficulty in proving this lemma is that the later steps are dependent on the previous steps. In the proof we show the dependency is bounded and in fact we can treat them as independent.

Lemma B.3 (Restated)**.** *Under the induction hypothesis (up to update step $\tilde{x}^{(t+1)} := A(y^{(t)})^{*2}$ at iteration t), we have for $i \in [t]$,*

$$|\langle v^{(i)}, w^{(t)} \rangle| \le O\left(t^3 \frac{(\Delta'_{t-1})^4}{(\delta'_{t-1})^4 \delta_t^2} (\log d) \frac{\sqrt{k}}{d} \right) = \tilde{O}\left(\frac{\sqrt{k}}{d} \right).$$

Proof: Recall $w^{(t)} = y_{-1}^{(t)} * y_{-1}^{(t)}$, and $y_{-1}^{(t)}$ is specified in (B.3). We now expand the Hadamard product in $w^{(t)}$ and bound all the resulting $O(t^2)$ terms.

The first type of terms has the form $\langle v^{(i)}, P_{\perp_{W^{[i_1-1]}}} w^{(i_1)} * P_{\perp_{W^{[i_2-1]}}} w^{(i_2)} \rangle$, which can be bounded as

$$\begin{aligned} \langle v^{(i)}, P_{\perp_{W^{[i_1-1]}}} w^{(i_1)} * P_{\perp_{W^{[i_2-1]}}} w^{(i_2)} \rangle &\le k \cdot \|v^{(i)}\|_\infty \cdot \|P_{\perp_{W^{[i_1-1]}}} w^{(i_1)} * P_{\perp_{W^{[i_2-1]}}} w^{(i_2)}\|_\infty \\ &\le 2k \frac{\log d}{\sqrt{d}} \frac{(\Delta'_{t-1})^2}{d^2}, \end{aligned}$$

where $\|v^{(i)}\|_\infty$ is bounded by Lemma B.4, and ℓ_∞ norm of other vector is bounded by induction Hypothesis 2. In addition, the corresponding coefficient is bounded by (see (B.5), and note that both $i_1, i_2 < t$)

$$\frac{4(\Delta'_{t-1})^2}{(\delta'_{t-1})^4} \frac{d^2}{k}.$$

Hence, the total contribution from such terms is bounded by

$$8t^2 \frac{(\Delta'_{t-1})^4}{(\delta'_{t-1})^4} \frac{\log d}{\sqrt{d}}. \tag{B.13a}$$

The second type of terms has the form $\langle v^{(i)}, P_{\perp_{W^{[i_1-1]}}} w^{(i_1)} * v^{(i_2)} \rangle = \langle v^{(i)} * v^{(i_2)}, P_{\perp_{W^{[i_1-1]}}} w^{(i_1)} \rangle$, which can be bounded as

$$\|P_{\perp_{W^{[i_1-1]}}} w^{(i_1)}\|_\infty \cdot \|v^{(i)} * v^{(i_2)}\|_1 \le \|P_{\perp_{W^{[i_1-1]}}} w^{(i_1)}\|_\infty \cdot \frac{\|v^{(i)}\|^2 + \|v^{(i_2)}\|^2}{2} \le 4\Delta'_{t-1} \frac{k}{d^2},$$

where the last inequality is concluded from Hypotheses 2 and 4. In addition, the corresponding coefficient is bounded by (see (B.4) and (B.5), and note that both $i_1, i_2 < t$)

$$\frac{2\Delta'_{t-1}}{(\delta'_{t-1})^2\delta_{t-1}}\frac{d}{\sqrt{k}}.$$

Hence, the total contribution from such terms is bounded by

$$8t^2\frac{(\Delta'_{t-1})^2}{\delta_{t-1}(\delta'_{t-1})^2}\frac{\sqrt{k}}{d}. \tag{B.13b}$$

The third type of terms has the form $\langle v^{(i)}, v^{(i_1)} * v^{(i_2)}\rangle$, with coefficient bounded by $1/\delta_{t-1}^2$ (see (B.4)). For bounding these inner products, we need to use the fact that they are random Gaussian vectors, however the main difficulty is that they are correlated (if $i > j$, then the subspace that $v^{(i)}$ is in that depends on $v^{(j)}$). To resolve this difficulty, we treat $v^{(i)} \in \mathbb{R}^{k-1}$ as projection of $n^{(i)} \in \mathbb{R}^{k-1}$ into subspace orthogonal to $W^{[t-1]}$, where $n^{(i)}$'s are *independent* Gaussian vectors in the full $k-1$ dimensional space. Independent of the ordering of i, i_1, i_2, we have with high probability

$$\langle n^{(i)}, n^{(i_1)} * n^{(i_2)}\rangle \le O\left(\frac{\sqrt{k}}{d\sqrt{d}}\right),$$

since it is a sum of $k-1$ independent mean-0 entries each with variance $\frac{1}{d^3}$. On the other hand, from Hypothesis 4, we have $\mathbb{E}[\|v^{(i)}\|^2] \le 4\frac{k}{d}$, and since vector $n^{(i)} - v^{(i)}$ is in the subspace $W^{[t-1]}$ with dimension t, we have

$$\mathbb{E}[\|n^{(i)} - v^{(i)}\|^2] \le O\left(\frac{t}{k}\right)\cdot\frac{4k}{d} = O\left(\frac{t}{d}\right),$$

and therefore, we have with high probability $\|n^{(i)} - v^{(i)}\| \le O(\sqrt{t/d})$ for all $i \in [t-1]$. Using this, the difference between $\langle n^{(i)}, n^{(i_1)} * n^{(i_2)}\rangle$ and $\langle v^{(i)}, v^{(i_1)} * v^{(i_2)}\rangle$ can be bounded as

$$|\langle n^{(i)}, n^{(i_1)} * n^{(i_2)}\rangle - \langle v^{(i)}, v^{(i_1)} * v^{(i_2)}\rangle| \le O\left((\log k)t\frac{\sqrt{k}}{d\sqrt{d}}\right),$$

where the right hand side is the bound on the dominant term in the expansion of difference as

$$\begin{aligned}|\langle n^{(i)}, (n^{(i_1)} - v^{(i_1)}) * (n^{(i_2)} - v^{(i_2)})\rangle| &\leq \|n^{(i)}\| \cdot \|(n^{(i_1)} - v^{(i_1)}) * (n^{(i_2)} - v^{(i_2)})\| \\ &\leq O\left((\log k)\sqrt{\frac{k}{d}}\right) \cdot O\left(\frac{t}{d}\right) \\ &= O\left((\log k) t \frac{\sqrt{k}}{d\sqrt{d}}\right).\end{aligned}$$

Here, the first inequality is the Cauchy-Schwartz, and the second inequality is from bound on the norm of random Gaussian vector $n^{(i)}$, and the bound on the norm of difference vectors $n^{(i_1)} - v^{(i_1)}$ stated earlier. Hence, the total contribution from such terms is bounded by

$$O\left(t^3 \frac{\log k}{\delta_{t-1}^2} \frac{\sqrt{k}}{d\sqrt{d}}\right). \tag{B.13c}$$

Taking the sum of all the terms in (B.13a)-(B.13c) gives the desired bound.

□

B.3 Additional Arguments for Noise Analysis

Proof of Lemma 2.7: We prove this by an induction argument.

Basis of induction: For $t = 1$, $x^{(1)}$ is the initialization vector and thus, $\xi^{(1)} = 0$. Hence, the proposed bound holds for the basis of induction $t = 1$.

Inductive step: Assuming the inductive hypothesis holds for step t, we prove it also holds for step $t+1$. We have

$$\begin{aligned} x^{(t+1)} + \xi^{(t+1)} &= \text{Norm}\left(\hat{T}(x^{(t)} + \xi^{(t)}, x^{(t)} + \xi^{(t)}, I)\right) \\ &= \text{Norm}\left(T(x^{(t)}, x^{(t)}, I) + 2T(x^{(t)}, \xi^{(t)}, I) + T(\xi^{(t)}, \xi^{(t)}, I) + E(\hat{x}^{(t)}, \hat{x}^{(t)}, I)\right). \end{aligned} \tag{B.14}$$

The first term $T(x^{(t)}, x^{(t)}, I)$ corresponds to the main signal; recall that $x^{(t+1)} = \text{Norm}(T(x^{(t)}, x^{(t)}, I))$ in the noiseless setting, where the unnormalized version $\tilde{x}^{(t+1)} := T(x^{(t)}, x^{(t)}, I)$ has norm at least $\tilde{\Omega}(\sqrt{k}/d)$ which is argued in the induction argument for Hypothesis 1. We now bound the desired property of noise terms in the above expansion.

For the second term, we break it into two terms as

$$2T(x^{(t)}, \xi^{(t)}, I) = 2\langle x^{(t)}, a_1\rangle\langle \xi^{(t)}, a_1\rangle a_1 + 2T'(x^{(t)}, \xi^{(t)}, I) =: p + q,$$

where $T' := \sum_{j>1} a_j \otimes a_j \otimes a_j$. Here $p := 2\langle x^{(t)}, a_1\rangle\langle \xi^{(t)}, a_1\rangle a_1$ corresponds to the multilinear form from first component of T, and $q := 2T'(x^{(t)}, \xi^{(t)}, I)$ corresponds to the multilinear form from the rest of components.

For q, we apply Lemma B.6. Note that since $\|x^{(t)}\|_{B^*} \le \tilde{O}(1/\sqrt{d})$, we get an extra $1/\sqrt{d}$ factor in the bound provided by Lemma B.6, and therefore we have

$$\|q\|_2 \le \tilde{O}(\epsilon d^{\beta 2^{t-1}}\sqrt{k}/d),$$

where we also used the induction hypothesis $\|\xi^{(t)}\| \le \tilde{O}(\epsilon d^{\beta 2^{t-1}})$.

For p, we have

$$\|p\| = 2|\langle x^{(t)}, a_1\rangle| \cdot |\langle \xi^{(t)}, a_1\rangle| \le \tilde{O}\left(\epsilon d^{\beta 2^t}\sqrt{k}/d\right),$$

where the inequality is from the signal and noise induction hypotheses; see Equation (2.26) for the signal induction hypothesis.

The third term $T(\xi^{(t)}, \xi^{(t)}, I)$ has ℓ_2 norm bounded as

$$\|T(\xi^{(t)}, \xi^{(t)}, I)\| \leq \|T\|\|\xi^{(t)}\|^2 \leq \tilde{O}(d^{\beta 2^t}\epsilon^2) \leq \tilde{O}(\epsilon d^{\beta 2^t}\sqrt{k}/d),$$

where the first inequality uses the sub-multiplicative property, and the second inequality exploits the bounded norm of random tensor T as $\|T\| \leq O(1)$, and the induction hypothesis in t-th step. The final inequality uses the assumption $\epsilon < o(\sqrt{k}/d)$ in the lemma.

The fourth term $E(\hat{x}^{(t)}, \hat{x}^{(t)}, I)$ has ℓ_2 norm bounded by

$$\|E(\hat{x}^{(t)}, \hat{x}^{(t)}, I)\| \leq \|E\|\|\hat{x}^{(t)}\|^2 \leq \epsilon\sqrt{k}/d,$$

where we use the sub-multiplicative property in the first inequality, and the assumption on the norm of error tensor E in the lemma, and the fact that $\|\hat{x}^{(t)}\| = 1$ are exploited in the second inequality.

Summarizing the above calculations on different terms of the update in (B.14), the signal plus noise vector before normalization is

$$T(x^{(t)}, x^{(t)}, I) + 2T(x^{(t)}, \xi^{(t)}, I) + T(\xi^{(t)}, \xi^{(t)}, I) + E(\hat{x}^{(t)}, \hat{x}^{(t)}, I) =: \alpha x^{(t+1)} + z,$$

where α is a coefficient which is lower bounded as $\alpha \geq \tilde{\Omega}(\sqrt{k}/d)$. The vector z also satisfies

$$\|z\| \leq \tilde{O}(\epsilon d^{\beta 2^t}\sqrt{k}/d), \tag{B.15}$$

which is derived by combining the bounds we argued on the second, third and fourth terms.

Note that until the very last step we always have $d^{\beta 2^t} \leq o(d/\sqrt{k})$ (otherwise we are constantly close to the true component, and we are done). In this case the norm of z is negligible compared to α since $\|z\| \leq o(\alpha)$, and thus, the normalization factor is equal to $\|\alpha x^{(t+1)} + z\| = \alpha(1 \pm o(1))$. Therefore, after the normalization, we have the noise vector $\xi^{(t+1)} = \alpha' x^{(t+1)} + \beta z$, where $|\alpha'| \leq \|z\|/\alpha \leq o(1)$ and $|\beta| \leq 2/\alpha \leq \tilde{O}(d/\sqrt{k})$, hence we know $\|\xi^{(t+1)}\| \leq \tilde{O}(\epsilon d^{\beta 2^t})$.

For the last step of the induction, the norm of $T(x^{(t)}, x^{(t)}, I)$ is also larger (it has norm $d^{\beta 2^t} k/d^2$, which is larger than $\sqrt{k}/d$ for the last step). Since $\epsilon < o(\sqrt{k}/d)$ we still know the noise is negligible.

□

Lemma on the property of $\|\cdot\|_*$ norm defined in Definition 2.2:

Lemma B.6. *Consider a random tensor $T = \sum_{j\in[k]} a_j \otimes a_j \otimes a_j$ where a_j's are zero-mean random Gaussian with expected unit norm. Let $A \in \mathbb{R}^{d\times k}$ be the matrix $[a_1, \ldots, a_k]$, $T' = \sum_{j>1} a_j \otimes a_j \otimes a_j$ and $B \in \mathbb{R}^{d\times(k-1)}$ be the matrix $[a_2, a_3, \ldots, a_k]$. Then for any vectors u, v such that $\|u\|_{B^*} \le 1$ and $\|v\|_2 \le 1$, with high probability we have*

$$\|T'(u, v, I)\|_2 \le \tilde{O}\left(\sqrt{k/d}\right).$$

Proof: We prove this lemma along similar ideas provided in the proof of Anandkumar et al. [21, Claim 1]. Let η_j's be independent random ± 1 variables with $\Pr[\eta_j = 1] = 1/2$. We rewrite tensor T' as

$$T' = \sum_{j>1} \eta_j a_j \otimes a_j \otimes a_j. \tag{B.16}$$

Since a_j's are zero-mean random Gaussian vectors, we have $\eta_j a_j \sim a_j$, and thus, the new T' has the same distribution as the original one. We now first sample vectors a_j's, and this already makes the norm $\|\cdot\|_{B*}$ well-defined. In addition, the value of η_j's does not change the singular values of A or B. Also note that since a_j's are zero-mean random Gaussian vectors with expected norm 1, they also satisfy with high probability the incoherence condition such that $|\langle a_i, a_j\rangle| \le \tilde{O}(1/\sqrt{d})$ for all $i \neq j$. Thus, we condition on all these fixed events, and the only remaining random variables are just the η_j's.

The proposed statement in the lemma is equivalent to bounding

$$\sup_{\|u\|_{B*}=1, \|v\|=\|w\|=1} \left|T'(u, v, w)\right|.$$

In order to bound it, we provide an ϵ-net argument. We construct an ϵ-net such that for any vector $u \in \mathbb{R}^d$ with unit $\|\cdot\|_{B^*}$ norm, there is a vector u' in the net such that $\|B^\top(u-u')\| \le 1/k^2$. We also construct standard ε-net for vectors $u, w \in \mathbb{R}^d$ with unit ℓ_2 norm. By standard construction, this ε-net has size $\exp(O(d\log d))$. We now show that for all u in ϵ-net with unit $\|\cdot\|_{B*}$ norm, and all v, w in ϵ-net with unit ℓ_2 norm, the desired bound $|T'(u,v,w)| \le \tilde{O}\left(\sqrt{k/d}\right)$ holds with high probability. Then for the other vectors (u,v,w) not in the ε-net, the result follows from their closest points in the net.

Now for a fixed triple (u,v,w) in the ε-net, we have

$$T'(u,v,w) = \sum_{j>1} \eta_j \langle u, a_j\rangle\langle v, a_j\rangle\langle w, a_j\rangle,$$

which is a sum of independent random variables; recall that the randomness is from η_j's, and a_j's are already sampled and thus they are fixed here. We partition the above sum into *large* and *small* terms as $T'(u,v,w) = S_L + S_{L^c}$ such that the summation S_L is the sum of large terms including terms in set

$$L := \left\{ j \in \{2,3,\dots,k\} : |\langle v, a_j\rangle| \ge \log d/\sqrt{d} \vee |\langle w, a_j\rangle| \ge \log d/\sqrt{d} \right\},$$

and the rest are the small terms forming S_{L^c}. Note that $|\langle u, a_j\rangle| \le 1$ since $\|u\|_{B*} = 1$.

Bounding $|S_{L^c}|$: Since the variables are bounded in this summation corresponding to small terms, we use Bernstein's inequality, and thus with probability at least $1-\delta$, we have $|S_{L^c}| \le \frac{\sqrt{k\log 1/\delta}\cdot \mathrm{polylog}\, d}{d}$ for the fixed point in the ε-net. By choosing small enough $\delta = \exp(-Cd\log d)$ (where C is large enough constant), we can apply the union bound on the ε-net, and conclude that for all the vectors in the net, $|S_{L^c}|$ is smaller than $\tilde{O}(\sqrt{k/d})$ with high probability.

Bounding $|S_L|$: Since the columns of matrix B are random Gaussian vectors, it satisfies the RIP property with high probability (see Remark 3 in Anandkumar et al. [21] for the precise definition of RIP), and thus by the definition of RIP and Lemma 3 in Anandkumar et al. [21], we have $\|B_L\| \le 2$ where B_L is the sub-columns of matrix B specified by set L.

We now have

$$|S_L| \leq \sum_{j \in L} |\langle u, a_j \rangle| \cdot |\langle v, a_j \rangle| \cdot |\langle w, a_j \rangle| \leq \sum_{j \in L} |\langle v, a_j \rangle| \cdot |\langle w, a_j \rangle| \leq \left\| B_L^\top v \right\| \cdot \left\| B_L^\top w \right\| \leq 4,$$

where the second step uses the fact that $|\langle u, a_j \rangle| \leq 1$, the third step exploits Cauchy-Schwartz inequality, and the last step uses bound $\|B_L\| \leq 2$. Notice that matrix B is already sampled before we do the ε-net argument, and therefore, we do not need to do union bound over all u, v, w for this event.

Since we assume the overcomplete regime $k \geq d$, the bound on $|S_{L^c}|$ is dominant which finishes the proof.

□

Appendix C

Proofs for Learning Overcomplete Latent Variable Models

C.1 Proof of Learning Theorems

The semi-supervised and unsupervised learning results for each latent variable model are proved by combining the corresponding tensor concentration bound proposed in Section 3.4 and the convergence guarantees of the tensor decomposition algorithm provided in Chapter 2.

Proof of Theorem 3.7: The result is proved by applying the tensor concentration bound in Theorem 3.4 to the local convergence result of Algorithm 1 recapped in Theorem 2.4. Note that in the high noise regime $\zeta^2 = \Theta(1)$, the term $\zeta^3\sqrt{\frac{d}{n}}$ in Theorem 3.4 is dominant, and in the low noise regime $\zeta^2 = \Theta\left(\frac{1}{d}\right)$, the term $\zeta\sqrt{w_{\max}\frac{d}{n}}$ in Theorem 3.4 is dominant.

Note that the sub-Gaussian property of conditional observed distributions is used to provide the labeled sample complexity. Since the distribution of observed variables given hidden state is sub-Gaussian with covariance matrix $\zeta^2 I$ as in model $\mathcal{S}$ described in Section 3.3.1, we have the following

concentration bound where with probability at least $1-\delta$, the empirical estimate $\widehat{a}_j^{(0)}$ satisfies

$$\left\|\widehat{a}_j^{(0)} - a_j\right\| \leq C_1 \sqrt{\frac{\zeta^2 d \log(1/\delta)}{m_j}}, \quad j \in [k],$$

for some constant $C_1 > 0$. □

Proof of Theorem 3.8: The result is proved by applying the tensor concentration bound in Theorem 3.4 to the global convergence result of Algorithm 1 recapped in Theorem 2.5. The dominant error bounds in Theorem 3.4 are the same as what stated in the proof of Theorem 3.7. □

Proof of Theorem 3.12: The learning results for the sparse ICA are proved similar to the ICA case, with the difference that the sparse ICA concentration bound in Theorem 3.6 is exploited here. □

Proof of Theorem 3.13: Given linear model $x = Ah$, the 4th order observed moment is expanded as

$$\mathbb{E}\left[x^{\otimes 4}\right] = \mathbb{E}\left[h^{\otimes 4}\right]\left(A^\top, A^\top, A^\top, A^\top\right), \tag{C.1}$$

where the multilinear notation defined in (1.2) is exploited.

Expanding $\mathbb{E}\left[h^{\otimes 4}\right]$, and treating $\sum_{i\in[k]} \mathbb{E}[h_i^4]\ e_i^{\otimes 4}$ as the main signal, the remaining term is

$$R := \sum_{i \neq j} \mathbb{E}[h_i^2 h_j^2]\ e_i^{\otimes 2} \otimes e_j^{\otimes 2},$$

where we also exploited the assumption that the expectation of terms involving odd powers of h_i are zero. Then, from (C.1), the spectral norm of perturbation tensor is bounded as

$$\|\Psi\| := \left\|R\left(A^\top, A^\top, A^\top, A^\top\right)\right\| = \left\|\sum_{i\neq j} \mathbb{E}[h_i^2 h_j^2]\ a_i^{\otimes 2} \otimes a_j^{\otimes 2}\right\| \leq \tau \|A\|^4,$$

where we used $\mathbb{E}[h_i^2 h_j^2] \leq \tau$ in the last inequality. Imposing condition $\|\Psi\| \leq \tilde{O}\left(w_{\min}/d\right)$, and then applying Theorem 2.5, the result is proved. Note that $w_{\min} := \min_{i\in[k]} \mathbb{E}[h_i^4] = \beta s/k$. □

C.2 Proof of Tensor Concentration Bounds

In this section, we provide the proof of tensor concentration bounds for different latent variable models including multiview linear mixtures model, ICA and sparse ICA. In order to get polynomial sample complexity bounds for unlabeled samples in semi-supervised and unsupervised learning results, it is usually enough to treat the tensor as a vector/matrix and apply appropriate vector/matrix concentration bounds such as Bernstein bounds. However, these bounds can be significantly improved in many cases by considering the concentration property of the tensor spectral norm directly.

C.2.1 Multiview linear mixtures model

In this section, we prove the tensor concentration result for the multiview linear mixtures model provided in Theorem 3.4.

Proof of Theorem 3.4: Expanding the difference $\hat{T} - \tilde{T}$, we have

$$\hat{T} - \tilde{T} = \frac{1}{n}\zeta^3 d^{1.5} \sum_{i\in[n]} \varepsilon_A^i \otimes \varepsilon_B^i \otimes \varepsilon_C^i \tag{C.2a}$$

$$+ \frac{1}{n}\zeta^2 d \sum_{i\in[n]} \left(a_{h_i} \otimes \varepsilon_B^i \otimes \varepsilon_C^i + \varepsilon_A^i \otimes b_{h_i} \otimes \varepsilon_C^i + \varepsilon_A^i \otimes \varepsilon_B^i \otimes c_{h_i}\right) \tag{C.2b}$$

$$+ \frac{1}{n}\zeta\sqrt{d} \sum_{i\in[n]} \left(a_{h_i} \otimes b_{h_i} \otimes \varepsilon_C^i + a_{h_i} \otimes \varepsilon_B^i \otimes c_{h_i} + \varepsilon_A^i \otimes b_{h_i} \otimes c_{h_i}\right). \tag{C.2c}$$

There are three types of terms in the above difference which are bounded separately in Claims 12-14 in Section C.2.1.2. Combining the results of claims, the theorem follows directly.

□

C.2.1.1 Basic definitions and lemmata

In the proof of the claims in Section C.2.1.2, we extensively apply two different types of partitioning as follows.

Definition C.1 (Small and large terms)**.** *Consider matrices* $E_A := [\varepsilon_A^1, \varepsilon_A^2, \ldots, \varepsilon_A^n] \in \mathbb{R}^{d\times n}$*, and* E_B *and* E_C *which are similarly defined. For any set of vectors* u*,* v *and* w*, the set of columns* $[n]$ *are partitioned into* 2 *sets called sets of* small *and* large *terms according to the value of inner products* $\langle u, \varepsilon_A^i\rangle$*,* $\langle v, \varepsilon_B^i\rangle$ *and* $\langle w, \varepsilon_C^i\rangle$ *as follows. The set of small values denoted by* $L^c \subseteq [n]$ *is defined as*

$$L^c := \left\{ i \in [n] : |\langle u, \varepsilon_A^i\rangle| < \frac{10\log d}{\sqrt{d}} \wedge |\langle v, \varepsilon_B^i\rangle| < \frac{10\log d}{\sqrt{d}} \wedge |\langle w, \varepsilon_C^i\rangle| < \frac{10\log d}{\sqrt{d}} \right\},$$

and the rest of columns belong to the set of large values denoted by $L \subseteq [n]$*.*

Note that when necessary, the above partitioning is similarly applied to one or two matrices.

Lemma C.1. *Suppose matrix* $E := [\varepsilon^1, \varepsilon^2, \ldots, \varepsilon^n] \in \mathbb{R}^{d\times n}$ *satisfies the RIP property (RIP). For a vector* $u \in \mathbb{R}^d$*, let set* $L \subseteq [n]$ *denote the set of columns of* E *corresponding to large inner products* $\langle u, \varepsilon^i\rangle$ *as defined in Definition C.1, i.e.,*

$$L := \left\{ i \in [n] : |\langle u, \varepsilon^i\rangle| \geq \frac{10\log d}{\sqrt{d}} \right\}.$$

Then, the size of set L *is bounded as*

$$|L| \leq \frac{d}{25\log^2 d}. \tag{C.3}$$

Proof: It can be shown by a contradiction argument assuming $|L| > \frac{d}{25\log^2 d}$. Consider submatrix $E[L]$ (matrix E with columns restricted to set L). We have

$$\|E\|^2 \geq \left\| E[L]^\top u \right\|^2 = \sum_{i\in L} \langle u, \varepsilon^i\rangle^2 \geq |L| \frac{100\log^2 d}{d} > 4,$$

where the first inequality is from the definition of large terms for which $|\langle u, \varepsilon^i \rangle| > 10 \log d/\sqrt{d}$, and the second inequality is from contradiction assumption on $|L|$. This contradicts with the RIP property that $\|E[L]\| \leq 2$, and therefore the bound in (C.3) holds. □

The above partitioning into small and large sets is good when all we care about is the inner-products between a fixed vector and the noise vectors. However, when we are also interested in the inner-products between a fixed vector and columns of A, B, C, it is often not tight enough, and in order to get a tight bound, we propose the following finer partitioning.

Definition C.2 (Buckets and constrained vectors)**.** *Consider matrix $C := [c_1, c_2, \ldots, c_k] \in \mathbb{R}^{d \times k}$, and let $t := \left\lceil \log_2 \sqrt{d} \right\rceil$. For any unit vector w, the set of columns $[k]$ are partitioned into $t+1$ buckets according to the value of inner products $\langle c_j, w \rangle$ as*

$$K_0 := \left\{ j \in [k] : |\langle c_j, w \rangle| \leq \frac{1}{\sqrt{d}} \right\},$$
$$K_l := \left\{ j \in [k] : |\langle c_j, w \rangle| \in \left(\frac{2^{l-1}}{\sqrt{d}}, \frac{2^l}{\sqrt{d}} \right] \right\}, \quad l \in [t].$$

Furthermore, the constrained vector $z^l \in \mathbb{R}^k, l \in \{0, 1, 2, \ldots, t\}$, corresponds to the inner products in bucket l as

$$z^l_j := \begin{cases} \langle c_j, w \rangle, & j \in K_l, \\ 0, & j \notin K_l. \end{cases}$$

One advantage of bucketing (which is not applicable to the small and large partitioning in the previous definition) is that buckets with large value has a smaller ε-net. This exploits the additional property of matrices with bounded $2 \to 3$ norm.

Lemma C.2. *Consider matrix $C := [c_1, c_2, \ldots, c_k] \in \mathbb{R}^{d \times k}$ where the columns have unit norm, and $\|C^\top\|_{2 \to 3} = O(1)$. For a vector w with unit norm, consider the buckets on columns of matrix C defined in Definition C.2. For constrained vector $z^l, l \in [t]$, let $p_l := 2^{l-1}$. Then, we have*

- *z^l has at most $O\left(\frac{d^{3/2}}{p_l^3}\right)$ nonzero entries.*

- *There is an ε-net of size* $\exp\left(O\left(\frac{d^{3/2}}{p_l^3}\left(\log k + \log\frac{1}{\varepsilon}\right)\right)\right)$ *for* z^l.

Proof: For the first part, we know the number of non-zero entries in z^l is $|K_l|$. For any unit vector w, we have

$$O(1) \geq \left\|C^\top w\right\|_3^3 \geq \sum_{j\in K_l} |\langle w, c_j\rangle|^3 \geq |K_l| \left(\frac{p_l}{\sqrt{d}}\right)^3,$$

which implies the desired bound on $|K_l|$.

Let $q_l := O\big(\frac{d^{3/2}}{p_l^3}\big)$ be the maximum number of nonzero entries in z^l. First enumerate the support of z^l. There are $\binom{k}{q_l}$ possibilities for the location of q_l nonzero entries in z^l which is bounded as

$$\binom{k}{q_l} \leq \left(e\frac{k}{q_l}\right)^{q_l} \leq e^{O(q_l \log k)}.$$

For a given support, take an ε-net for all vectors in that support which has size

$$e^{O\left(q_l \log\left(\frac{1}{\varepsilon}\right)\right)}.$$

The union of these ε-nets is a valid ε-net for z^l of the desired size. This finishes the proof of second claim.

□

A similar (but stronger) lemma can be proved for RIP matrices:

Lemma C.3. *Consider matrix $E := [\varepsilon^1, \varepsilon^2, \ldots, \varepsilon^n] \in \mathbb{R}^{d\times n}$ where the columns have unit norm, and it satisfies RIP property (RIP). For a vector w with unit norm, consider the buckets on columns of matrix E defined in Definition C.2. For constrained vector z^l, let $p_l := 2^{l-1}$. Then, for $l > 4\log\log d$ we have*

- z^l *has at most* $O\left(\frac{d}{p_l^2}\right)$ *nonzero entries.*
- *There is an ε-net of size* $\exp\left(O\left(\frac{d}{p_l^2}\left(\log n + \log\frac{1}{\varepsilon}\right)\right)\right)$ *for* z^l.

Proof: The first claim follows from the same argument as in Lemma C.1. The ε-net is constructed in the same way as in the previous lemma. □

C.2.1.2 Proof of claims

In this section, we separately bound different error terms (C.2a)-(C.2c). Among all the terms, the terms like (C.2c) is most difficult to bound (intuitively because terms like b_{h_i} are not "as random" as terms like ε_A^i). In fact, the proof for the term (C.2c) can be adapted to bound all the other terms. Here for clarity we start from the simplest term (C.2a), and point out new ideas in the proofs of (C.2b) and (C.2c).

Claim 12 (Bounding norm of (C.2a))**.** *With high probability over $\varepsilon_A^i, \varepsilon_B^i, \varepsilon_C^i$'s and h_i's, we have*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} \varepsilon_A^i \otimes \varepsilon_B^i \otimes \varepsilon_C^i \right\| \leq \tilde{O}\left(\frac{1}{n} + \frac{1}{d\sqrt{n}} \right).$$

Proof: Let

$$T_1 := \frac{1}{n} \sum_{i=1}^{n} \varepsilon_A^i \otimes \varepsilon_B^i \otimes \varepsilon_C^i.$$

Rewrite the tensor as

$$T_1 = \frac{1}{n} \sum_{i=1}^{n} \eta_i \varepsilon_A^i \otimes \varepsilon_B^i \otimes \varepsilon_C^i, \tag{C.4}$$

where η_i's are independent random ± 1 variables with $\Pr[\eta_i = 1] = 1/2$. Clearly, T_1 has the same distribution as the original term, because of the symmetry in error vectors implying e.g. $\eta_i \varepsilon_A^i \sim \varepsilon_A^i$. We first sample the vectors $\varepsilon_A^i, \varepsilon_B^i, \varepsilon_C^i$, and therefore, the remaining random variables are just the η_i's.

The goal is to bound norm of T_1 in (C.4) which is defined as

$$\|T_1\| := \sup_{\|u\|=\|v\|=\|w\|=1} |T_1(u,v,w)| = \sup_{\|u\|=\|v\|=\|w\|=1} \left| \frac{1}{n} \sum_{i=1}^{n} \eta_i \langle u, \varepsilon_A^i \rangle \langle v, \varepsilon_B^i \rangle \langle w, \varepsilon_C^i \rangle \right|. \tag{C.5}$$

In order to bound the above, we provide an ε-net argument. Construct an ε-net for vectors u, v and w with $\varepsilon = 1/n^2$. By standard construction, size of the ε-net is $e^{O(d \log n)}$. First, for any fixed triple (u, v, w), we bound $|T_1(u, v, w)|$ where $T_1(u, v, w)$ is a sum of independent variables. As introduced in Definition C.1, we partition the sum into *large* and *small* terms as

$$T_1(u, v, w) = \frac{1}{n} \sum_{i=1}^{n} \eta_i \langle u, \varepsilon_A^i \rangle \langle v, \varepsilon_B^i \rangle \langle w, \varepsilon_C^i \rangle := S_L + S_{L^c},$$

where S_{L^c} is the sum of *small* terms consisting of terms satisfying

$$\left\{ |\langle u, \varepsilon_A^i \rangle| < \frac{10 \log d}{\sqrt{d}} \wedge |\langle v, \varepsilon_B^i \rangle| < \frac{10 \log d}{\sqrt{d}} \wedge |\langle w, \varepsilon_C^i \rangle| < \frac{10 \log d}{\sqrt{d}} \right\},$$

and S_L is the sum of *large* terms including all the other terms.

Bounding $|S_{L^c}|$: The sum S_{L^c} is just a weighted sum of η_i's, and the Bernstein's Inequality is exploited to bound it. Each term in the summation is bounded as

$$\left| \frac{1}{n} \langle u, \varepsilon_A^i \rangle \langle v, \varepsilon_B^i \rangle \langle w, \varepsilon_C^i \rangle \right| \leq O\left(\frac{\log^3 d}{n d^{3/2}} \right),$$

where the bound on the small terms is exploited. The variance term is also bounded as

$$O\left(\frac{\log^6 d}{n d^3} \right).$$

Applying Bernstein's inequality, with probability at least $1 - e^{-Cd \log n}$ (where C is a large enough constant), the sum of small terms $|S_{L^c}|$ is bounded by $\tilde{O}\left(\frac{1}{d\sqrt{n}} \right)$.

Bounding $|S_L|$: From RIP property (RIP), we know that noise matrices $E_A := [\varepsilon_A^1, \ldots, \varepsilon_A^n]$, $E_B := [\varepsilon_B^1, \ldots, \varepsilon_B^n]$ and $E_C := [\varepsilon_C^1, \ldots, \varepsilon_C^n]$ satisfy the weak RIP condition with high probability such that for any subset of $O\left(\frac{d}{\log^2 d} \right)$ number of columns, the spectral norm of matrices restricted to those columns is bounded by 2. Let L denote the set of large terms in the proposed partitioning, and $E_A[L]$, $E_B[L]$ and $E_C[L]$ be the matrices E_A, E_B and E_C restricted to the columns indexed by L.

Applying Lemma C.1, we have

$$|L| \leq \frac{3d}{25\log^2 d}.$$

Note that an additional factor 3 shows up here since the set of small terms is defined as the intersection of 3 sets comparing to what proved in Lemma C.1. Therefore, RIP property of E_A, E_B and E_C implies that $E_A[L]$, $E_B[L]$ and $E_C[L]$ have spectral norm bounded by 2. Now applying triangle inequality, we have

$$|S_L| \leq \frac{1}{n}\sum_{i\in L} |\langle u, \varepsilon_A^i\rangle|\cdot|\langle v, \varepsilon_B^i\rangle|\cdot|\langle w, \varepsilon_C^i\rangle| \leq \frac{1}{n}\sum_{i\in L} |\langle u, \varepsilon_A^i\rangle|\cdot|\langle v, \varepsilon_B^i\rangle| \leq \frac{1}{n}\left\|E_A[L]^\top u\right\|\cdot\left\|E_B[L]^\top v\right\| \leq \frac{4}{n},$$

where the second step uses the fact that $|\langle w, \varepsilon_C^i\rangle| \leq 1$, the third step exploits Cauchy-Schwartz inequality, and the last step uses bounds $\|E_A[L]\| \leq 2$ and $\|E_B[L]\| \leq 2$. Notice the three matrices are already sampled before we do the ε-net argument, and therefore, we do not need to do union bound over all u, v, w for this event.

At this point, we have bounds on $|S_L|$ and $|S_{L^c}|$ for a fixed triple (u, v, w) in the ε-net. By applying union bound on all vectors in the ε-net, the bound holds for every triple (u, v, w) in the ε-net. The argument for other (u, v, w)'s which are not in the ε-net follows from their closest triples in the ε-net. □

Claim 13 (Bounding norm of (C.2*b*))**.** *With high probability over $\varepsilon_A^i, \varepsilon_B^i$'s and h_i's, we have*

$$\left\|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_A^i \otimes \varepsilon_B^i \otimes c_{h_i}\right\| \leq \tilde{O}\left(\frac{1}{n} + \sqrt{\frac{w_{\max}}{n\sqrt{d}}}\right).$$

Proof: The proof is similar to the previous claim. Let

$$T_2 = \frac{1}{n}\sum_{i=1}^{n} \eta_i \varepsilon_A^i \otimes \varepsilon_B^i \otimes c_{h_i},$$

where η_i's are independent random ± 1 variables with $\Pr[\eta_i = 1] = 1/2$. Similar to the previous claim, we first sample the vectors $\varepsilon_A^i, \varepsilon_B^i$ and h_i's, and therefore, the remaining random variables are just the η_i's. Assume the matrices E_A, E_B satisfy the RIP property, and the number of times

$h_i = j$ for $j \in [k]$ is bounded by $[nw_{\min}/2, 2nw_{\max}]$. All the events happen with high probability when $n \geq \tilde{\Omega}(1/w_{\min})$ and $n \leq \text{poly}(k)$.

The goal is to bound $\|T_2\|$. We construct an ε-net for vectors u and v with $\varepsilon = 1/n^2$. First, for any fixed pair (u, v), we bound $\|T_2(u, v, I)\|$ where $T_2(u, v, I)$ is a sum of independent zero mean vectors. As introduced in Definition C.1, consider partitioning on columns of E_A and E_B as

$$T_2(u, v, I) = \frac{1}{n} \sum_{i=1}^{n} \eta_i \langle u, \varepsilon_A^i \rangle \langle v, \varepsilon_B^i \rangle c_{h_i} = S_L + S_{L^c},$$

where S_{L^c} is the sum of *small* terms consisting of terms satisfying

$$\left\{ |\langle u, \varepsilon_A^i \rangle| < \frac{10 \log d}{\sqrt{d}} \ \wedge \ |\langle v, \varepsilon_B^i \rangle| < \frac{10 \log d}{\sqrt{d}} \right\},$$

and S_L is the sum of *large* terms including all the other terms.

Bounding $\|S_L\|$: This is bounded in a similar way to the argument for bounding S_L in the previous claim. From RIP property (RIP), we know that noise matrices $E_A := [\varepsilon_A^1, \ldots, \varepsilon_A^n]$ and $E_B := [\varepsilon_B^1, \ldots, \varepsilon_B^n]$ satisfy the weak RIP condition with high probability. Let L be the set of large terms in the proposed partitioning, and $E_A[L]$, $E_B[L]$ be the matrices E_A, E_B restricted to the columns indexed by L. Applying Lemma C.1, we have

$$|L| \leq \frac{2d}{25 \log^2 d}.$$

Therefore, RIP property of E_A and E_B implies that $E_A[L]$ and $E_B[L]$ have spectral norm bounded by 2. Applying triangle inequality, we have

$$\|S_L\| \leq \frac{1}{n} \sum_{i \in L} |\langle u, \varepsilon_A^i \rangle| \cdot |\langle v, \varepsilon_B^i \rangle| \leq \frac{1}{n} \left\| E_A[L]^\top u \right\| \cdot \left\| E_B[L]^\top v \right\| \leq \frac{4}{n},$$

where Cauchy-Schwartz inequality is exploited in the second inequality, and the bounds $\|E_A[L]\| \leq 2$ and $\|E_B[L]\| \leq 2$ are used in the last inequality. Notice the two matrices are already sampled before we do the ε-net argument, and therefore, we do not need to do union bound over all u, v for this event.

Bounding $\|S_{L^c}\|$: Similar to how we bounded $|S_{L^c}|$ in the previous claim by applying Bernstein's inequality, it is tempting to apply vector Bernstein's inequality here. However, vector Bernstein's inequality does not utilize the fact that the matrix $C^\top$ has small $2 \to 3$ norm, and results in a suboptimal bound. Here, we try to exploit this additional property to to get a better bound.

Let L^c denote the set of small terms in the proposed partitioning on columns of E_A and E_B. Then, we have

$$\langle S_{L^c}, w\rangle = \frac{1}{n}\sum_{i\in L^c} \eta_i \langle u, \varepsilon_A^i\rangle\langle v, \varepsilon_B^i\rangle\langle w, c_{h_i}\rangle.$$

Now, we try to bound the above inner product $\langle S_{L^c}, w\rangle$ by considering an ε-net on w as well (Note that the ε-nets on u and v are already considered). To do that we partition the inner products $\langle c_j, w\rangle$ into $t+1$ buckets ($t := \lceil \log_2 \sqrt{d}\rceil$) as defined in Definition C.2 where

$$K_0 := \left\{ j \in [k] : |\langle c_j, w\rangle| \le \frac{1}{\sqrt{d}} \right\},$$
$$K_l := \left\{ j \in [k] : |\langle c_j, w\rangle| \in \left(\frac{2^{l-1}}{\sqrt{d}}, \frac{2^l}{\sqrt{d}} \right] \right\}, \quad l \in [t].$$

Let Q_l denote the sum of all terms that fall into bucket K_l as

$$Q_l := \frac{1}{n}\sum_{i\in L^c, h_i \in K_l} \eta_i \langle u, \varepsilon_A^i\rangle\langle v, \varepsilon_B^i\rangle\langle w, c_{h_i}\rangle. \tag{C.6}$$

Note that by construction of buckets, we have

$$\langle S_{L^c}, w\rangle = \sum_{l=0}^{t} Q_l.$$

There are only $O(\log d)$ terms in this summation, and therefore, it suffices to show each term Q_l is small.

For Q_0, it is a weighted sum of η_i's with weights bounded by $\tilde{O}(1/d^{3/2})$, so the situation is exactly the same as Claim 12.

For $Q_l, l \in [t]$, the argument is as follows. Let $p_l := 2^{l-1}$. Applying Lemma C.2, we have

$$|K_l| \leq O\left(\frac{d^{3/2}}{p_l^3}\right).$$

As stated in the beginning of proof, each hidden state $h_i \in [k]$ appears in at most $O(2nw_{\max})$ samples w.h.p. Hence, the total number of terms in the summation form (C.6) for Q_l is w.h.p. bounded as

$$|\{i \in [n] : h_i \in K_l\}| \leq O\left(nw_{\max}\frac{d^{3/2}}{p_l^3}\right).$$

Now the sum Q_l in (C.6) is a weighted sum of η_i's and the Bernstein's inequality is exploited to bound it. Each term in the summation is bounded as

$$\tilde{O}\left(\frac{p_l}{nd^{3/2}}\right),$$

where the bound on the small terms and the bound on terms in bucket K_l are exploited. The variance term is also bounded as

$$O\left(\frac{w_{\max}}{np_l d^{3/2}}\right).$$

Applying Bernstein's inequality, with probability at least $1 - e^{-Cd\log n}$ for large enough constant C, we have (notice below that $p_l \leq O(\sqrt{d})$)

$$Q_l \leq \tilde{O}\left(\frac{p_l}{\sqrt{d}n} + \sqrt{\frac{w_{\max}}{np_l\sqrt{d}}}\right) \leq \tilde{O}\left(\frac{1}{n} + \sqrt{\frac{w_{\max}}{n\sqrt{d}}}\right).$$

At this point, we have bounds on $\|S_L\|$ and $\|S_{L^c}\|$ for a fixed pair of vectors (u, v) in the ε-net. By applying union bound on all vectors in the ε-net, the bound holds for every pair (u, v) in the ε-net. The argument for other (u, v)'s which are not in the ε-net follows from their closest pairs in the ε-net. □

Now we are ready to bound the last term (C.2c).

Claim 14 (Bounding norm of (C.2c))**.** *With high probability over ε_A^i's and h_i's, we have*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} \varepsilon_A^i \otimes b_{h_i} \otimes c_{h_i} \right\| \leq \tilde{O}\left(\frac{1}{n} + \sqrt{\frac{w_{\max}}{n}} \right).$$

Proof: Again, rewrite the tensor as

$$T_3 = \frac{1}{n} \sum_{i=1}^{n} \eta_i \varepsilon_A^i \otimes b_{h_i} \otimes c_{h_i}, \tag{C.7}$$

where η_i's are independent random ± 1 variables with $\Pr[\eta_i = 1] = 1/2$. First sample ε_A^i and h_i's, and therefore, the remaining random variables are just the η_i's. In addition, assume $E_A := [\varepsilon_A^1, \varepsilon_A^2, \ldots, \varepsilon_A^n]$ satisfies the RIP property (RIP) and each $h_i \in [k]$ appears between $nw_{\min}/2$ and $2nw_{\max}$ times where both events happen with high probability.

The goal is to bound norm of T_3 in (C.7) which is defined as

$$\|T_3\| := \sup_{\|u\|=\|v\|=\|w\|=1} |T_3(u,v,w)| = \sup_{\|u\|=\|v\|=\|w\|=1} \left| \frac{1}{n} \sum_{i=1}^{n} \langle u, \varepsilon_A^i \rangle \langle v, b_{h_i} \rangle \langle w, c_{h_i} \rangle \right|. \tag{C.8}$$

In order to bound the above, we provide an ε-net argument similar to what we did for bounding S_{L^c} in the previous claim with the difference that here we apply bucketing to all three matrices E_A, B and C. First, for any fixed triple (u, v, w), we partition the inner products in (C.8) into buckets as defined in Definition C.2. Let K_l^a, K_l^b and K_l^c denote the bucketing of matrices E_A, B and C, respectively.

In addition, we merge the buckets K_0^a, $K_1^a, \ldots, K_{4\log\log d}^a$ into K_0^a. This means K_0^a now contains all i's with inner product

$$|\langle \varepsilon_A^i, u \rangle| \leq \frac{16 \log d}{\sqrt{d}},$$

and K_l^a's for $1 \leq l \leq 4 \log\log d$ are empty. Let

$$J_{l_1,l_2,l_3} := \left\{ i \in [n] : i \in K_{l_1}^a \wedge h_i \in K_{l_2}^b \wedge h_i \in K_{l_3}^c \right\},$$

and Q_{l_1,l_2,l_3} be the sum of terms in summation (C.8) on this set, i.e.,

$$Q_{l_1,l_2,l_3} := \frac{1}{n} \sum_{i \in J_{l_1,l_2,l_3}} \langle u, \varepsilon_A^i \rangle \langle v, b_{h_i} \rangle \langle w, c_{h_i} \rangle. \tag{C.9}$$

Note that by construction of buckets, the summation in (C.8) is expanded as

$$\frac{1}{n} \sum_{i=1}^{n} \langle u, \varepsilon_A^i \rangle \langle v, b_{h_i} \rangle \langle w, c_{h_i} \rangle = \sum_{l_1,l_2,l_3=0}^{t} Q_{l_1,l_2,l_3}.$$

There are only $O(t^3) = O(\log^3 d)$ terms in this summation, and therefore, it suffices to show each term Q_{l_1,l_2,l_3} is small.

For $Q_{0,0,0}$, it is a weighted sum of η_i's with weights bounded by $\tilde{O}(1/d^{3/2})$, and therefore, it follows from the same arguments as Claim 12.

For Q_{l_1,l_2,l_3} with $\max\{l_1, l_2, l_3\} > 0$, let $p_l := 2^{\max\{l_1,l_2,l_3\}-1}$. By Lemma C.2 and Lemma C.3, the total number of terms in the summation form (C.9) for Q_{l_1,l_2,l_3} is w.h.p. bounded as

$$|J_{l_1,l_2,l_3}| \leq O\left(n w_{\max} \frac{d^{3/2}}{p_l^3} \right),$$

and there exists an ε-net of size

$$\exp\left(O\left(\frac{d^{3/2}}{p_l^3} \log n \right) \right)$$

with $\varepsilon < 1/n^2$. For every u, v, w in the ε-net, this term $n \cdot Q_{l_1,l_2,l_3}$ is a weighted sum of η_i's, and the Bernstein's inequality is exploited to bound it. Each term in the summation is bounded as $\frac{8p_l^3}{d^{3/2}}$, where the bound on the terms in buckets are exploited. The variance term is also bounded as

$$O\left(n w_{\max} \frac{p_l^3}{d^{3/2}} \right).$$

Applying Bernstein's inequality, with probability at least $1 - \exp\left(-C \frac{d^{3/2}}{p_l^3} \log n\right)$ for large enough constant C, we have

$$n Q_{l_1,l_2,l_3} \leq \tilde{O}\left(1 + \sqrt{n w_{\max}}\right).$$

Taking the union bound over all triples in ε-net, this bound holds for all such triples. For u, v, w which are not in the ε-net, the bound follows from the closest point in the ε-net.

□

C.2.2 ICA

In this section, we prove the tensor concentration result for the ICA model provided in Theorem 3.5.

Recall the 4th order modified moment tensor in equation (3.5) as

$$M_4 := \mathbb{E}[x \otimes x \otimes x \otimes x] - T,$$

where $T \in \mathbb{R}^{d \times d \times d \times d}$ is the fourth order tensor with

$$T_{i_1,i_2,i_3,i_4} := \mathbb{E}[x_{i_1} x_{i_2}]\mathbb{E}[x_{i_3} x_{i_4}] + \mathbb{E}[x_{i_1} x_{i_3}]\mathbb{E}[x_{i_2} x_{i_4}] + \mathbb{E}[x_{i_1} x_{i_4}]\mathbb{E}[x_{i_2} x_{i_3}], \quad i_1, i_2, i_3, i_4 \in [d].$$

Let $\widehat{M_4}$ be the empirical estimate of M_4 given n samples.

Proof of Theorem 3.5: Let $W := \frac{1}{n} \sum_{i=1}^{n} x^i (x^i)^\top$, and therefore, the empirical estimate of T is given by

$$\widehat{T}_{i_1,i_2,i_3,i_4} = W_{i_1,i_2} W_{i_3,i_4} + W_{i_1,i_3} W_{i_2,i_4} + W_{i_1,i_4} W_{i_2,i_3}. \tag{C.10}$$

Then, the empirical estimate of M_4 is given by

$$\widehat{M_4} = \frac{1}{n} \sum_{i=1}^{n} (x^i)^{\otimes 4} - \widehat{T}.$$

The proof directly follows from Claims 16 and 17, which bound the perturbation of the two terms separately. Claim 16 bounds the 4th order term perturbation $\mathbb{E}[x^{\otimes 4}] - \frac{1}{n} \sum_{i=1}^{n} (x^i)^{\otimes 4}$, and Claim 17 bounds the 2nd order term perturbation $T - \widehat{T}$. □

C.2.2.1 Proof of claims

Before bounding the 4-th order term we first give the following claim which bounds a sum of subgaussian variables raised to the 4-th power.

Claim 15. *Suppose $h_i, i \in [n]$, are independent q-subgaussian random variables. Then, for any $d \geq 1$, with probability at least $1 - e^{-\omega(d \log n)}$ we have*

$$\left| \frac{1}{n} \sum_{i=1}^{n} \left(h_i^4 - \mathbb{E}\left[h_i^4\right] \right) \right| \leq \tilde{O}\left(\frac{q^4 d^2}{n} + \sqrt{\frac{q^8 d}{n}} \right).$$

(Notice that here d is intended to be the dimension in later applications. However, for this claim we can choose d to be an arbitrary real number that is at least 1.)

Proof: We prove

$$\Pr\left[\frac{1}{n} \left| \sum_{i=1}^{n} h_i^4 - \mathrm{med}\Big(\sum_{i=1}^{n} h_i^4 \Big) \right| \leq \tilde{O}\left(\frac{q^4 d^2}{n} + \sqrt{\frac{q^8 d}{n}} \right) \right] \geq 1 - e^{-\omega(d \log n)}, \tag{C.11}$$

where $\mathrm{med}(\cdot)$ is the median of the distribution. By doing simple integration (for d from 1 to ∞), this concentration bound implies

$$\left| \mathbb{E}\left[\sum_{i=1}^{n} h_i^4 \right] - \mathrm{med}\Big(\sum_{i=1}^{n} h_i^4 \Big) \right| \leq \tilde{O}\left(\frac{q^4}{\sqrt{n}} \right).$$

Therefore, when $d \geq 1$ the difference between mean and median is negligible, and we get the desired bound in the claim.

In order to prove the deviation bound from the median in (C.11), we use the standard symmetrization argument: it is enough to take two independent sample sets $\{h_1, h_2, \ldots, h_n\}$ and $\{\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_n\}$ with the same distribution, and bound $\left| \frac{1}{n} \sum_{i \in [n]} \left(h_i^4 - \tilde{h}_i^4 \right) \right|$. In order to bound the sum, we rewrite it in the form

$$Q = \frac{1}{n} \sum_{i \in [n]} \eta_i |h_i^4 - \tilde{h}_i^4|,$$

where η_i's are independent random ± 1 variables with $\Pr[\eta_i = 1] = 1/2$.

Now we partition the terms in the summation for Q into multiple buckets according to the magnitude of $\left|h_i^4 - \tilde{h}_i^4\right|$. Let $t := \lceil \log_2 d^2 + C' \log_2 \log_2 n \rceil$ (where C' is a large enough constant). Then the buckets are defined as

$$\begin{aligned} K_0 &:= \left\{ i \in [n] : |h_i^4 - \tilde{h}_i^4| \le q^4 \right\}, \\ K_l &:= \left\{ i \in [n] : |h_i^4 - \tilde{h}_i^4| \in \left(2^{l-1} q^4, 2^l q^4 \right] \right\}, \quad l \in [t], \\ K_{t+1} &:= \left\{ i \in [n] : |h_i^4 - \tilde{h}_i^4| > 2^t q^4 \right\}. \end{aligned}$$

Let Q_l denote the sum of all terms that fall into bucket K_l as

$$Q_l := \frac{1}{n} \sum_{i \in [n], i \in K_l} |h_i^4 - \tilde{h}_i^4| \eta_i. \tag{C.12}$$

Note that by construction of buckets, the original summation $Q = \sum_{l=0}^{t+1} Q_l$. There are only $O(\log d)$ terms in this summation, and therefore, it suffices to show each term Q_l is small.

Note that since h_i's and $\tilde{h}_i$'s are q-subgaussian random variables, we have

$$\begin{aligned} \Pr\left[|h_i^4 - \tilde{h}_i^4| \ge \lambda q^4 \right] &\le \Pr\left[h_i^4 \ge \lambda q^4 / 2 \right] + \Pr\left[\tilde{h}_i^4 \ge \lambda q^4 / 2 \right] \\ &= 2 \Pr\left[|h_i| \ge (\lambda q^4 / 2)^{1/4} \right] \\ &\le 4 \exp\left(- \frac{\sqrt{\lambda}}{2\sqrt{2}} \right), \end{aligned} \tag{C.13}$$

where the last inequality uses q-subgaussian property.

For $Q_l, 0 \le l \le 2 \log \log n$, we apply Bernstein's inequality directly. Each term in the summation for Q_l is bounded as $\tilde{O}(q^4/n)$, and the variance term is also bounded as $\tilde{O}(q^8/n)$. By applying Bernstein's inequality, with probability at least $1 - e^{-\omega(d \log n)}$, we have

$$Q_l \le \tilde{O}\left(\frac{q^4 d}{n} + \sqrt{\frac{q^8 d}{n}} \right), \quad 0 \le l \le 2 \log \log n.$$

For $Q_l, 2\log\log n < l \le t$, we first bound the number of terms in bucket K_l. From (C.13), we have

$$\Pr\left[|K_l| \ge \tilde{\Omega}\left(d2^{-l/2}\right)\right] \le e^{-\omega(d\log n)}.$$

Each term in the summation Q_l is bounded by $2^l q^4/n$, and therefore, by applying triangle inequality we have with probability at least $1 - e^{-\omega(d\log n)}$

$$Q_l \le \tilde{O}\left(d2^{-l/2}\right)\frac{2^l q^4}{n} \le \tilde{O}\left(\frac{q^4 d 2^{l/2}}{n}\right) \le \tilde{O}\left(\frac{q^4 d^2}{n}\right), \quad 2\log\log n < l \le t.$$

Here the last inequality uses the fact that $l \le t$, which implies $2^{l/2} = \tilde{O}(d)$.

For the last term Q_{t+1}, again from (C.13), we have with probability at least $1 - e^{-\omega(d\log n)}$, there is only one term in the sum and that particular term is smaller than $\tilde{O}(q^4 d^2/n)$.

Now by union bound, with probability at least $1 - e^{-\omega(d\log n)}$ all the terms are bounded by $\tilde{O}\big(q^4 d^2/n + \sqrt{q^8 d/n}\big)$, which implies the summation Q is also bounded by

$$\tilde{O}\left(\frac{q^4 d^2}{n} + \sqrt{\frac{q^8 d}{n}}\right).$$

□

Now we are ready to bound the 4-th order term perturbation $\mathbb{E}[x^{\otimes 4}] - \frac{1}{n}\sum_{i=1}^n (x^i)^{\otimes 4}$.

Claim 16. *Suppose $\|A\| \le O(\sqrt{k/d})$ and the entries of $h \in \mathbb{R}^k$ are independent subgaussian variables with $\mathbb{E}[h_j^2] = 1$. Given n samples $x^i = Ah^i, i \in [n]$, we have with high probability*

$$\left\|\frac{1}{n}\sum_{i\in[n]}(x^i)^{\otimes 4} - \mathbb{E}[x^{\otimes 4}]\right\| \le \tilde{O}\left(\frac{k^2}{n} + \sqrt{\frac{k^4}{d^3 n}}\right).$$

Proof: The desired spectral norm in the lemma is defined as

$$\sup_{\|u\|=1}\left|\frac{1}{n}\sum_{i\in[n]}\langle u, x^i\rangle^4 - \mathbb{E}[\langle u, x\rangle^4]\right|.$$

In order to bound it, we provide an ε-net argument. Construct an ε-net for vectors u in the unit ball $\mathcal{S}^{d-1}$ with $\varepsilon = 1/n^2$. By standard construction, size of the ε-net is $e^{O(d \log n)}$. For any fixed u in the ε-net, let $v := A^\top u$. Since $x^i = Ah^i, i \in [n]$, we have $\langle u, x^i \rangle = \langle v, h^i \rangle$. Therefore, for any fixed u (and the corresponding v) in the ε-net, we would like to bound

$$Q := \frac{1}{n} \sum_{i \in [n]} \left(\langle v, h^i \rangle^4 - \mathbb{E}[\langle v, h^i \rangle^4] \right).$$

Since h^i's have independent subgaussian entries, we know that $\langle v, h^i \rangle$ is $\|v\|$-subgaussian. On the other hand, we have

$$\|v\| \leq \|A\| \|u\| = O(\sqrt{k/d}),$$

and therefore, $\langle v, h^i \rangle$ is a $O(\sqrt{k/d})$-subgaussian random variable. By Claim 15, with probability at least $1 - e^{-Cd \log n}$ (for large enough constant C) we have

$$|Q| \leq \tilde{O}\left(\frac{k^2}{n} + \sqrt{\frac{k^4}{d^3 n}} \right).$$

By applying union bound on all vectors in the ε-net, the bound holds for every vector u in the ε-net. The argument for other u's which are not in the ε-net follows from their closest vectors in the ε-net.

□

The 2nd order term T in (3.6) is sum of three terms, each of which is an outer-product of two matrices. Hence, it is good enough to apply a matrix concentration for bounding this term.

Claim 17. *Suppose $\|A\| \leq O(\sqrt{k/d})$ and the entries of $h \in \mathbb{R}^k$ are independent subgaussian variables with $\mathbb{E}[h_j^2] = 1$. Given n samples $x^i = Ah^i, i \in [n]$, for T in* (3.6) *and the empirical estimate $\widehat{T}$ in* (C.10)*, if $n \geq d$, we have with high probability*

$$\|\widehat{T} - T\| \leq \tilde{O}\left(\sqrt{\frac{k^4}{d^3 n}} \right).$$

Proof: Recall $W := \frac{1}{n}\sum_{i=1}^{n} x^i (x^i)^\top$. We prove the result for the first term

$$\widehat{T}_1[i_1, i_2, i_3, i_4] = W_{i_1,i_2} W_{i_3,i_4},$$

or equivalently $\widehat{T}_1 = W \otimes W$. The analysis for the other two terms follow similarly from symmetry.

Let $T_1 = \mathbb{E}[xx^\top] \otimes \mathbb{E}[xx^\top] = \mathbb{E}[W] \otimes \mathbb{E}[W]$. We have

$$\widehat{T}_1 - T_1 = (W - \mathbb{E}[W]) \otimes \mathbb{E}[W] + \mathbb{E}[W] \otimes (W - \mathbb{E}[W]) + (W - \mathbb{E}[W]) \otimes (W - \mathbb{E}[W]).$$

For any matrices A and B, we have $\|A \otimes B\| \leq \|A\|\|B\|$. Thus,

$$\|\widehat{T}_1 - T_1\| \leq 2\|W - \mathbb{E}[W]\| \cdot \|\mathbb{E}[W]\| + \|W - \mathbb{E}[W]\|^2. \tag{C.14}$$

We bound $\|W - \mathbb{E}[W]\|$ by Matrix Bernstein's inequality. For applying Matrix Bernstein's inequality, we need a bound on the norm of each term in the summation form of W, i.e., bound on $\|x^i (x^i)^\top\|$ which holds almost surely. Therefore, we apply the Bernstein's inequality on the bounded version of W as

$$W' := \frac{1}{n}\sum_{i=1}^{n} x^i (x^i)^\top \mathbf{1}_{\|x^i\| \leq O(\sqrt{k}\log n)},$$

where $\mathbf{1}_{\|x^i\| \leq O(\sqrt{k}\log n)}$ is an indicator variable. Since $x = Ah$ and entries of h are subgaussian, the indicator variables are 1 with probability $1 - n^{-\log n}$. Therefore, W and W' are equal with high probability at it suffices to apply Matrix Bernstein's bound on W'.

For the summation W', the norm of each term is bounded by $\tilde{O}(k/n)$, and for the variance term, we have

$$\mathbb{E}\left[W'(W')^\top\right] = \frac{1}{n}\mathbb{E}\left[\|x^i\|^2 x^i (x^i)^\top \mathbf{1}_{\|x^i\| \leq O(\sqrt{k}\log n)}\right] \preceq \frac{1}{n}\tilde{O}(k)\mathbb{E}\left[x^i (x^i)^\top\right] = \frac{1}{n}\tilde{O}(k) AA^\top.$$

Since $\|A\| \leq O(\sqrt{k/d})$, it is concluded that the variance is bounded by $\tilde{O}(k^2/dn)$. Therefore, Matrix Bernstein's inequality implies that with probability at least $1 - d/n$,

$$\|W' - \mathbb{E}[W']\| \leq \tilde{O}\left(\frac{k}{n} + \frac{k}{\sqrt{dn}}\right).$$

Since W is equal to W' with high probability and $\|\mathbb{E}[W] - \mathbb{E}[W']\|$ is negligible, we also have $\|W - \mathbb{E}[W]\| \leq \tilde{O}(k/\sqrt{dn})$ (when $n \geq d$).

On the other hand, $\mathbb{E}[W] = AA^\top$, and therefore, $\|\mathbb{E}[W]\| \leq k/d$. From (C.14), we have

$$\|\widehat{T}_1 - T_1\| \leq \tilde{O}\left(\sqrt{\frac{k^4}{d^3 n}}\right).$$

□

C.2.3 Sparse ICA

In this section, we prove the tensor concentration result for the sparse ICA model provided in Theorem 3.6. This is the sparse coding problem in the sparse ICA setting (where h_i's are independent and sparse). The proof can be generalized to the case when h_i's are negatively correlated or more generally when concentration bounds hold for h_i's.

The proof of Theorem 3.6 is similar to the proof of Theorem 3.5, where the 4th order term perturbation $\mathbb{E}[x^{\otimes 4}] - \frac{1}{n}\sum_{i=1}^n (x^i)^{\otimes 4}$, and the 2nd order term perturbation $T - \widehat{T}$ are separately bounded in the following two claims. First, we bound the perturbation of the 4th order term in the following claim. Note that this is the sparse version of Claim 16.

Claim 18. *Consider the sparse ICA model described in Theorem 3.6. Given n independent samples $x^i = Ah^i, i \in [n]$, we have with high probability*

$$\left\| \frac{1}{n}\sum_{i=1}^n (x^i)^{\otimes 4} - \mathbb{E}[x^{\otimes 4}] \right\| \leq \tilde{O}\left(\frac{s^2}{n} + \sqrt{\frac{s^4}{d^3 n}}\right).$$

Proof: The proof uses ideas from both Claims 13 and 15 . Without loss of generality, we assume $s/k < 1/2$. Otherwise, h_j's are 2-subgaussian, and therefore the dense case argument in Claim 16 implies the desired bound.

Let η_i's be independent random ± 1 variables with $\Pr[\eta_i = 1] = 1/2$. We equivalently bound

$$\left\| \frac{1}{n} \sum_{i=1}^{n} \eta_i \left((x^i)^{\otimes 4} - \mathbb{E}\left[(x^i)^{\otimes 4} \right] \right) \right\| := \sup_{\|u\|=1} \left| \frac{1}{n} \sum_{i \in [n]} \eta_i \left(\langle u, x^i \rangle^4 - \mathbb{E}[\langle u, x^i \rangle^4] \right) \right|.$$

In order to bound it, we provide an ε-net argument. Construct an ε-net for vectors u in the unit ball $\mathcal{S}^{d-1}$ with $\varepsilon = 1/n^2$. By standard construction, size of the ε-net is $e^{O(d \log n)}$. For any fixed u in the ε-net, let $v := A^\top u$. Since $x^i = Ah^i, i \in [n]$, we have $\langle u, x^i \rangle = \langle v, h^i \rangle$. Therefore, for any fixed u (and the corresponding v) in the ε-net, we would like to bound

$$\left| \frac{1}{n} \sum_{i \in [n]} \eta_i \left(\langle v, h^i \rangle^4 - \mathbb{E}[\langle v, h^i \rangle^4] \right) \right|. \tag{C.15}$$

Now, we follow the ideas of Claim 15, and apply the standard symmetrization trick: it is enough to take two independent sample sets $\{h^1, h^2, \ldots, h^n\}$ and $\{\tilde{h}^1, \tilde{h}^2, \ldots, \tilde{h}^n\}$ with the same distribution, and bound $\left| \frac{1}{n} \sum_{i \in [n]} \eta_i \left(\langle v, h^i \rangle^4 - \langle v, \tilde{h}^i \rangle^4 \right) \right|$ instead of (C.15). Note that the difference between mean and median here is negligible because our distributions have first and second moments polynomial in parameters, and strong exponential concentration. Therefore, for any vector u (and the corresponding v), we would like to bound the sum

$$\frac{1}{n} \sum_{i \in [n]} \eta_i \left| \langle v, h^i \rangle^4 - \langle v, \tilde{h}^i \rangle^4 \right|.$$

The techniques we use to prove bounds on sums of random variables $\sum_{i=1}^{n} \eta_i z_i$ (either Bernstein's inequality, or bounding the number of terms and then using triangle inequality) all works if we just know an *upper bound* of z_i. Therefore, we can equivalently bound

$$Q = \frac{1}{n} \sum_{i \in [n]} \eta_i \left(\langle v, h^i \rangle^4 + \langle v, \tilde{h}^i \rangle^4 \right),$$

where the subtraction is replaced with addition.

Now, we partition the entries of vector $v = A^\top u \in \mathbb{R}^k$ into different vectors v_l according to the magnitude of entries (this is very similar to Claim 13). In particular, we partition entries (inner products) $v_j = \langle u, a_j \rangle, j \in [k]$, into $t+1$ buckets ($t := \lceil \log_2 \sqrt{d} \rceil$) where (similar to Definition C.2)

$$K_0 := \left\{ j \in [k] : |\langle u, a_j \rangle| \leq \frac{1}{\sqrt{d}} \right\},$$
$$K_l := \left\{ j \in [k] : |\langle u, a_j \rangle| \in \left(\frac{2^{l-1}}{\sqrt{d}}, \frac{2^l}{\sqrt{d}} \right] \right\}, \quad l \in [t].$$

In addition, we merge the buckets $K_0, K_1, \ldots, K_{\frac{1}{2}\log\log d}$ into K_0. This means K_0 now contains all j's with inner product

$$|\langle u, a_j \rangle| \leq \frac{\sqrt{\log d}}{\sqrt{d}},$$

and K_l's for $1 \leq l \leq \frac{1}{2}\log\log d$ are empty. Now, let v_l denote the restriction of vector v to entries indexed by K_l, i.e.,

$$v_l(j) := \begin{cases} v(j), & j \in K_l, \\ 0, & j \notin K_l. \end{cases}$$

Let $p_l := 2^{l-1}$. By RIP property of matrix A, and exploiting Lemma C.3, the number of nonzero entries in v_l is bounded as

$$\|v_l\|_0 = |K_l| \leq O\left(\frac{d}{p_l^2} \right), \quad l > \frac{1}{2}\log\log d.$$

Exploiting the above partitioning, the term $\langle v, h^i \rangle^4$ in summation Q can be upper bounded as

$$\langle v, h^i \rangle^4 = \Big(\sum_{l=0}^{t} \langle v_l, h^i \rangle \Big)^4 \leq \Big((t+1) \sum_{l=0}^{t} \langle v_l, h^i \rangle^2 \Big)^2 \leq (t+1)^3 \sum_{l=0}^{t} \langle v_l, h^i \rangle^4,$$

where the equality is concluded from the fact that nonzero values of v_l's are derived from partitioning of values of v, and Cauchy-Schwartz inequality is exploited in the last two steps. Applying this

upper bound on Q, we would like to bound

$$Q' := \frac{1}{n} \sum_{i \in [n]} \eta_i (t+1)^3 \sum_{l=0}^{t} \left(\langle v_l, h^i \rangle^4 + \langle v_l, \tilde{h}^i \rangle^4 \right).$$

In order to bound Q', we break it into sum of $t+1$ terms as $Q' = \sum_{l=0}^{t} Q'_l$ where

$$Q'_l := \frac{1}{n} (t+1)^3 \sum_{i \in [n]} \eta_i \left(\langle v_l, h^i \rangle^4 + \langle v_l, \tilde{h}^i \rangle^4 \right).$$

All terms Q'_l can be bounded in the same way as Claim 15. Especially, directly from Claim 15, we have

$$Q'_0 \leq \tilde{O} \left(\frac{s^2}{n} + \sqrt{\frac{s^4}{d^3 n}} \right).$$

For the other terms $Q'_l, l > \frac{1}{2} \log \log d$, we need to analyze the tail behavior of $\langle v_l, h^i \rangle^4$. The tail behavior of this variable is affected by two phenomena: 1) the size of intersection of the supports of v_l and h^i, and 2) given the intersection, the tail behavior of

$$\langle v_l, h^i \rangle = \sum_{j \in [k] : s^i[j] = 1} v_l[j] g^i[j], \tag{C.16}$$

which is a sum of subgaussian random variables. Recall that $h^i[j] = s^i[j] g^i[j]$ where $s^i \in \mathbb{R}^k$ with i.i.d. Bernoulli random entries specifies the support of h^i.

The first part (the intersection of supports) can be bounded by Chernoff bound as

$$\Pr\Big[\sum_{j \in [k]} s^i[j] \geq (1+\delta) s \Big] \leq \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}} \right)^s.$$

The second part follows from subgaussian concentrations bounds. Let $\theta_l := \frac{2^l}{\sqrt{d}}$. For bucket K_l, and subsequently Q'_l where v_l has entries in the interval $(\theta_l/2, \theta_l]$, we discuss the tail behavior in two cases where $1/\theta_l^2 \geq s$ and $1/\theta_l^2 \leq s$.

Case 1 ($1/\theta_l^2 \geq s$): In this case, most of $\langle v_l, h^i \rangle^4$ are of size s^2/k^2 which is very small. For any $q \in \big[\sqrt{s/(k\theta_l^2)} \operatorname{polylog}(n), s \big]$, since the summation in (C.16) is $\sqrt{s}\theta_l$-subgaussian, with probability

at least $1 - e^{-\tilde{\Omega}(q)}$, we have

$$\langle v_l, h^i \rangle^4 \in \left(q^4\theta_l^4/2, q^4\theta_l^4\right].$$

Therefore in this range, with probability at least $1 - e^{-\tilde{\Omega}(1/\theta_l^2)}$, the summation Q_l' is bounded by

$$\frac{1}{n}\tilde{O}\left(\frac{q^4\theta_l^4}{\theta_l^2 q}\right) = \tilde{O}\left(\frac{q^3\theta_l^2}{n}\right) \leq \tilde{O}\left(\frac{s^2}{n}\right),$$

where the last inequality uses the fact that $\theta_l^2 \leq 1/s$.

For any $q \in \left(s, \sqrt{s/\theta_l^2}\log^2 n\right]$, since the summation in (C.16) is $\sqrt{s}\theta_l$-subgaussian, with probability at least $1 - e^{-\tilde{\Omega}(q^2/s)}$, we have

$$\langle v_l, h^i \rangle^4 \in \left(q^4\theta_l^4/2, q^4\theta_l^4\right].$$

Therefore in this range, with probability at least $1 - e^{-\tilde{\Omega}(1/\theta_l^2)}$, the summation Q_l' is bounded by

$$\frac{1}{n}\tilde{O}\left(q^4\theta_l^4\frac{1}{\theta_l^2 q^2/s}\right) = \tilde{O}\left(\frac{q^2\theta_l^2 s}{n}\right) \leq \tilde{O}\left(\frac{s^2}{n}\right),$$

where the last inequality uses the fact that $q^2 = \tilde{O}(s/\theta_l^2)$.

When $q > \sqrt{s/\theta_l^2}\log^2 n$, there are no term $\langle v_l, h^i \rangle^4$ in this range with high probability. Therefore, in the first case, by doing union bound Q_l' is always bounded by

$$\tilde{O}\left(\frac{s^2}{n}\right) + o\left(\frac{s^4}{d^3 n}\right).$$

Case 2 ($1/\theta_l^2 \leq s$): In this case, again most of $\langle v_l, h^i \rangle^4$ are of size s^2/k^2 which is very small. The only difference with case 1 is the two ranges where instead of being separated at s, they are separated at $1/\theta_l^2$ because there are at most $\tilde{O}(1/\theta_l^2)$ nonzero entries in v_l as shown earlier.

For any $q \in \left[\sqrt{s/(k\theta_l^2)}\,\mathrm{polylog}(n), 1/\theta_l^2\right]$, since the summation in (C.16) is $\sqrt{s}\theta_l$-subgaussian, with probability at least $1 - e^{-\tilde{\Omega}(q)}$, we have

$$\langle v_l, h^i \rangle^4 \in \left(q^4\theta_l^4/2, q^4\theta_l^4\right].$$

Therefore in this range, with probability at least $1 - e^{-\tilde{\Omega}(1/\theta_l^2)}$, the summation Q_l' is bounded by

$$\frac{1}{n}\tilde{O}\left(\frac{q^4\theta_l^4}{\theta_l^2 q}\right) = \tilde{O}\left(\frac{q^3\theta_l^2}{n}\right) \leq \tilde{O}\left(\frac{s^2}{n}\right),$$

where the last inequality uses the fact that $\theta_l^2 \leq 1/s$.

For any $q \in \left(1/\theta_l^2, \sqrt{s/\theta_l^2}\log^2 n\right]$, since the summation in (C.16) is $\sqrt{s}\theta_l$-subgaussian, with probability at least $1 - e^{-\tilde{\Omega}(q^2\theta_l^2)}$, we have

$$\langle v_l, h^i\rangle^4 \in \left(q^4\theta_l^4/2, q^4\theta_l^4\right].$$

Therefore in this range, with probability at least $1 - e^{-\tilde{\Omega}(1/\theta_l^2)}$, the summation Q_l' is bounded by

$$\frac{1}{n}\tilde{O}\left(q^4\theta_l^4\frac{1}{\theta_l^2 q^2\theta_l^2}\right) = \tilde{O}\left(\frac{q^2}{n}\right) \leq \tilde{O}\left(\frac{s^2}{n}\right),$$

where the last inequality uses the fact that $q^2 = \tilde{O}(s/\theta_l^2) \leq \tilde{O}(s^2)$.

When $q > \sqrt{s/\theta_l^2}\log^2 n$, there are no term $\langle v_l, h^i\rangle^4$ in this range with high probability. Therefore, in the second case, by doing union bound Q_l' is always bounded by

$$\tilde{O}\left(\frac{s^2}{n}\right) + o\left(\frac{s^4}{d^3 n}\right).$$

Combining the bounds on all terms finishes the proof. □

In the next claim we bound the perturbation of the 2nd order term T. Note that this is the sparse version of Claim 17.

Claim 19. *Consider the same sparse setting as in Theorem 3.6. Given n samples $x^i = Ah^i, i \in [n]$, where $\|A\| \leq O(\sqrt{k/d})$, for T in (3.6) and the empirical estimate $\widehat{T}$ in (C.10), if $n \geq d$, we have with high probability*

$$\|\widehat{T} - T\| \leq \tilde{O}\left(\sqrt{\frac{s^4}{d^3 n}}\right).$$

Proof: The proof is very similar to Claim 17. Recall $W := \frac{1}{n}\sum_{i=1}^{n} x^i(x^i)^\top$. We prove the result for the first term

$$\widehat{T}_1[i_1, i_2, i_3, i_4] = W_{i_1,i_2} W_{i_3,i_4},$$

or equivalently $\widehat{T}_1 = W \otimes W$. The analysis for the other two terms follow similarly from symmetry. As in (C.14), we have

$$\|\widehat{T}_1 - T_1\| \leq 2\|W - \mathbb{E}[W]\| \cdot \|\mathbb{E}[W]\| + \|W - \mathbb{E}[W]\|^2.$$

We bound $\|W - \mathbb{E}[W]\|$ by Matrix Bernstein's inequality. As in Claim 17, we first construct

$$W' = \frac{1}{n}\sum_{i=1}^{n} x^i(x^i)^\top \mathbf{1}_{\|x^i\| \leq O(\sqrt{s}\log n)},$$

where $\mathbf{1}_{\|x^i\| \leq O(\sqrt{s}\log n)}$ is an indicator variable. Since $x = Ah$ and entries of h are subgaussian, the indicator variables are 1 with probability $1 - n^{-\log n}$. Therefore W and W' are equal with high probability at it suffices to apply Matrix Bernstein's bound on W'.

For the summation W', the norm of each term is bounded by $\tilde{O}(s/n)$, and for the variance term, we have

$$\mathbb{E}\left[W'(W')^\top\right] = \frac{1}{n}\mathbb{E}\left[\|x^i\|^2 x^i(x^i)^\top \mathbf{1}_{\|x^i\| \leq O(\sqrt{s}\log n)}\right] \preceq \frac{1}{n}\tilde{O}(s)\mathbb{E}\left[x^i(x^i)^\top\right] = \frac{1}{n}\tilde{O}(s^2/k)AA^\top.$$

Since $\|A\| \leq O(\sqrt{k/d})$, it is concluded that the variance is bounded by $\tilde{O}(s^2/dn)$. Therefore, Matrix Bernstein's inequality implies that with probability at least $1 - d/n$,

$$\|W' - \mathbb{E}[W']\| \leq \tilde{O}\left(\frac{s}{n} + \frac{s}{\sqrt{dn}}\right).$$

Since W is equal to W' with high probability and $\|\mathbb{E}[W] - \mathbb{E}[W']\|$ is negligible, we also have $\|W - \mathbb{E}[W]\| \leq \tilde{O}(s/\sqrt{dn})$ (when $n \geq d$).

On the other hand, $\mathbb{E}[W] = \frac{s}{k} AA^\top$, and therefore, $\|\mathbb{E}[W]\| \leq s/d$. From (C.14), we have

$$\|\widehat{T}_1 - T_1\| \leq \tilde{O}\left(\sqrt{\frac{s^4}{d^3 n}}\right).$$

□

Appendix D

Proofs for Guaranteed Training of Neural Networks

D.1 Details of Tensor Decomposition Algorithm

The goal of tensor decomposition algorithm is to recover the rank-1 components of tensor; refer to Equation (1.5) for the notion of tensor rank and its rank-1 components. We exploit the tensor decomposition algorithm proposed in Section 2.3. In addition, the whitening procedure is also proposed here. For the sake of completeness, the power iteration and SVD initializations are again provided with the appropriate changes compared to Section 2.3. Figure D.1 depicts the flowchart of this method where the corresponding algorithms and procedures are also specified. Similarly, Algorithm 9 states the high-level steps of tensor decomposition algorithm. The whitening preprocessing is applied to orthogonalize the components of input tensor. Note that the convergence guarantees of tensor power iteration for orthogonal tensor decomposition have been developed in the literature [160, 18].

The tensorization step works as follows.

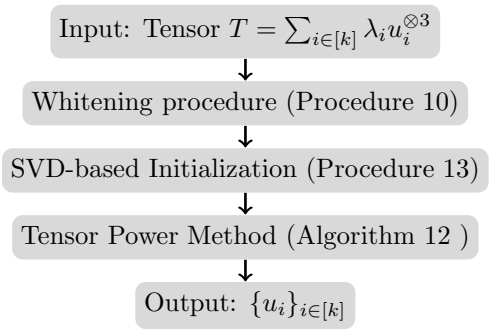


Figure D.1: Overview of tensor decomposition algorithm for third order tensor (without tensorization).

Algorithm 9 Tensor Decomposition Algorithm Setup

input symmetric tensor T.

if Whitening **then**
  Calculate T =Whiten(T); see Procedure 10.
else if Tensorizing **then**
  Tensorize the input tensor.
  Calculate T =Whiten(T); see Procedure 10.
for $j = 1$ to k **do**
  (v_j, μ_j, T) = tensor power decomposition(T); see Algorithm 12.
$(A_1)_j$ = Un-whiten$(v_j), j \in [k]$; see Procedure 11.
return $\{(A_1)_j\}_{j\in[k]}$.

Tensorization: The tensorizing step is applied when we want to decompose overcomplete tensors where the rank k is larger than the dimension d. For instance, for getting rank up to $k = O(d^2)$, we first form the 6th order input tensor with decomposition as

$$T = \sum_{j\in[k]} \lambda_j a_j^{\otimes 6} \in \bigotimes^6 \mathbb{R}^d.$$

Given T, we form the 3rd order tensor $\tilde{T} \in \bigotimes^3 \mathbb{R}^{d^2}$ which is the tensorization of T such that

$$\tilde{T}\big(i_2 + d(i_1 - 1), j_2 + d(j_1 - 1), l_2 + d(l_1 - 1)\big) := T(i_1, i_2, j_1, j_2, l_1, l_2). \tag{D.1}$$

This leads to $\tilde{T}$ having decomposition

$$\tilde{T} = \sum_{j\in[k]} \lambda_j (a_j \odot a_j)^{\otimes 3}.$$

Procedure 10 Whitening

input Tensor $T \in \mathbb{R}^{d\times d\times d}$.

1: Second order moment $M_2 \in \mathbb{R}^{d\times d}$ is constructed such that it has the same decompositon form as target tensor T (see Section D.2.1.1 for more discussions):

- Option 1: constructed using second order score function; see Equation (D.10).
- Option 2: computed as $M_2 := T(I, I, \theta) \in \mathbb{R}^{d\times d}$, where $\theta \sim \mathcal{N}(0, I_d)$ is a random standard Gaussian vector.

2: Compute the rank-k SVD, $M_2 = U\,\mathrm{Diag}(\gamma)U^\top$, where $U \in \mathbb{R}^{d\times k}$ and $\gamma \in \mathbb{R}^k$.

3: Compute the whitening matrix $W := U\,\mathrm{Diag}(\gamma^{-1/2}) \in \mathbb{R}^{d\times k}$.

4: **return** $T\,(W, W, W) \in \mathbb{R}^{k\times k\times k}$.

Procedure 11 Un-whitening

input Orthogonal rank-1 components $v_j \in \mathbb{R}^k, j \in [k]$.

1: Consider matrix M_2 which was exploited for whitening in Procedure 10, and let $\tilde{\lambda}_j, j \in [k]$ denote the corresponding coefficients as $M_2 = A_1\,\mathrm{Diag}(\tilde{\lambda})A_1^\top$; see (D.10).

2: Compute the rank-k SVD, $M_2 = U\,\mathrm{Diag}(\gamma)U^\top$, where $U \in \mathbb{R}^{d\times k}$ and $\gamma \in \mathbb{R}^k$.

3: Compute

$$(A_1)_j = \frac{1}{\sqrt{\tilde{\lambda}_j}} U\,\mathrm{Diag}(\gamma^{1/2})v_j, \quad j \in [k].$$

4: **return** $\{(A_1)_j\}_{j\in[k]}$.

We then apply the tensor decomposition algorithm to this new tensor $\tilde{T}$. This now clarifies why the full column rank condition is applied to the columns of $A \odot A = [a_1 \odot a_1 \cdots a_k \odot a_k]$. Similarly, we can perform higher order tensorizations leading to more overcomplete models by exploiting initial higher order tensor T; see also Remark 13.

Efficient implementation of tensor decomposition given samples: The main update steps in the tensor decomposition algorithm is the tensor power iteration for which a multilinear operation is performed on tensor T. However, the tensor is not available beforehand, and needs to be estimated using the samples (as in Algorithm 6 in the main text). Computing and storing the tensor can be enormously expensive for high-dimensional problems. But, it is essential to note that since we can form a factor form of tensor T using the samples and other parameters in the model, we can manipulate the samples directly to perform the power update as *multi-linear* operations without explicitly forming the tensor. This leads to efficient computational complexity. See [20] for details on these implicit update forms.

Algorithm 12 Robust tensor power method

input symmetric tensor $\tilde{T} \in \mathbb{R}^{d' \times d' \times d'}$, number of iterations N, number of initializations R.
output the estimated eigenvector/eigenvalue pair; the deflated tensor.

1: **for** $\tau = 1$ to R **do**
2: Initialize $\widehat{v}_0^{(\tau)}$ with SVD-based method in Procedure 13.
3: **for** $t = 1$ to N **do**
4: Compute power iteration update

$$\widehat{v}_t^{(\tau)} := \frac{\tilde{T}(I, \widehat{v}_{t-1}^{(\tau)}, \widehat{v}_{t-1}^{(\tau)})}{\|\tilde{T}(I, \widehat{v}_{t-1}^{(\tau)}, \widehat{v}_{t-1}^{(\tau)})\|} \tag{D.2}$$

5: Let $\tau^* := \arg\max_{\tau \in [R]} \{\tilde{T}(\widehat{v}_N^{(\tau)}, \widehat{v}_N^{(\tau)}, \widehat{v}_N^{(\tau)})\}$.
6: Do N power iteration updates (D.2) starting from $\widehat{v}_N^{(\tau^*)}$ to obtain $\widehat{v}$, and set $\hat{\mu} := \tilde{T}(\widehat{v}, \widehat{v}, \widehat{v})$.
7: **return** the estimated eigenvector/eigenvalue pair $(\widehat{v}, \hat{\mu})$; the deflated tensor $\tilde{T} - \hat{\mu} \cdot \hat{v}^{\otimes 3}$.

Procedure 13 SVD-based initialization

input Tensor $T \in \mathbb{R}^{d' \times d' \times d'}$.

1: **for** $\tau = 1$ to $\log(1/\hat{\delta})$ **do**
2: Draw a random standard Gaussian vector $\theta^{(\tau)} \sim \mathcal{N}(0, I_{d'})$.
3: Compute $u_1^{(\tau)}$ as the top left singular vector of $T(I, I, \theta^{(\tau)}) \in \mathbb{R}^{d' \times d'}$.
4: $\widehat{v}_0 \leftarrow \max_{\tau \in [\log(1/\hat{\delta})]} \left(u_1^{(\tau)}\right)_{\min}$.
5: **return** $\widehat{v}_0$.

D.2 Proof of Theorem 4.3

Proof of Theorem 4.3 includes three main pieces which is about arguing the recovery guarantees of three different parts of the algorithm: tensor decomposition, Fourier method, and linear regression. As the first piece, we show that the tensor decomposition algorithm for estimating weight matrix A_1 (see Algorithm 6 for the details) recovers it with the desired error. In the second part, we analyze the performance of Fourier technique for estimating bias vector b_1 (see Algorithm 6 and Procedure 7 for the details) proving the error in the recovery is small. Finally as the last step, the ridge regression is analyzed to ensure that the parameters of last layer of the neural network are well estimated leading to the estimation of overall function $\tilde{f}(x)$. We now provide the analysis of these three parts.

D.2.1 Tensor decomposition guarantees

We first provide a short proof for Lemma 4.1 which shows how the rank-1 components of third order tensor $\mathbb{E}\left[\tilde{y} \cdot \mathcal{P}_3(x)\right]$ are the columns of weight matrix A_1.

Proof of Lemma 4.1: It is shown by Janzamin et al. [102] that the score function yields differential operator such that for label-function $f(x) := \mathbb{E}[y|x]$, we have

$$\mathbb{E}[y \cdot \mathcal{S}_3(x)] = \mathbb{E}[\nabla_x^{(3)} f(x)].$$

Applying this property to the form of label function $f(x)$ in (4.4) denoted by $\tilde{f}(x)$, we have

$$\mathbb{E}\left[\tilde{y} \cdot \mathcal{P}_3(x)\right] = \mathbb{E}[\sigma'''(\cdot)(a_2, A_1^\top, A_1^\top, A_1^\top)],$$

where $\sigma'''(\cdot)$ denotes the third order derivative of element-wise function $\sigma(z) : \mathbb{R}^k \to \mathbb{R}^k$. More concretely, with slightly abuse of notation, $\sigma'''(z) \in \mathbb{R}^{k \times k \times k \times k}$ is a diagonal 4th order tensor with its j-th diagonal entry equal to $\frac{\partial^3 \sigma(z_j)}{\partial z_j^3} : \mathbb{R} \to \mathbb{R}$. Here two properties are used to compute the third order derivative $\nabla_x^{(3)} \tilde{f}(x)$ on the R.H.S. of above equation as follows. 1) We apply chain rule to take the derivatives which generates a new factor of A_1 for each derivative. Since we take 3rd order derivative, we have 3 factors of A_1. 2) The linearity of next layers leads to the derivatives from them being vanished, and thus, we only have the above term as the derivative. Expanding the above multilinear form finishes the proof; see (1.2) for the definition of multilinear form. □

We now provide the recovery guarantees of weight matrix A_1 through tensor decomposition as follows.

Lemma D.1. *Among the conditions for Theorem 4.3, consider the rank constraint on A_1, and the non-vanishing assumption on coefficients λ_j's. Let the whitening to be performed using empirical version of second order score function as specified in* (D.10)*, and assume the coefficients $\tilde{\lambda}_j$'s do*

not vanish. Suppose the sample complexity

$$n \geq \max\left\{\tilde{O}\left(\tilde{y}_{\max}^2 \mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right] \frac{\tilde{\lambda}_{\max}^4}{\tilde{\lambda}_{\min}^4} \frac{s_{\max}^2(A_1)}{\lambda_{\min}^2 \cdot s_{\min}^6(A_1)} \cdot \frac{1}{\tilde{\epsilon}_1^2}\right), \right. \tag{D.3}$$

$$\tilde{O}\left(\tilde{y}_{\max}^2 \cdot \mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right] \cdot \left(\frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}\right)^3 \frac{1}{\lambda_{\min}^2 \cdot s_{\min}^6(A_1)} \cdot k\right),$$

$$\left.\tilde{O}\left(\tilde{y}_{\max}^2 \cdot \frac{\mathbb{E}\left[\left\|\mathcal{S}_2(x)\mathcal{S}_2^\top(x)\right\|\right]^{3/2}}{\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]^{1/2}} \cdot \frac{1}{\tilde{\lambda}_{\min}^2 \cdot s_{\min}^3(A_1)}\right)\right\},$$

holds, where $M_3(x) \in \mathbb{R}^{d \times d^2}$ *denotes the matricization of score function tensor* $\mathcal{S}_3(x) \in \mathbb{R}^{d \times d \times d}$*; see* (1.1) *for the definition of matricization. Then the estimate* $\widehat{A}_1$ *by NN-LIFT Algorithm 6 satisfies w.h.p.*

$$\min_{z \in \{\pm 1\}} \|(A_1)_j - z \cdot (\widehat{A}_1)_j\| \leq \tilde{O}\left(\tilde{\epsilon}_1\right), \quad j \in [k],$$

where the recovery guarantee is up to the permutation of columns of A_1*.*

Remark 29 (Sign ambiguity). We observe that in addition to the permutation ambiguity in the recovery guarantees, there is also a sign ambiguity issue in recovering the columns of matrix A_1 through the decomposition of third order tensor in (4.16). This is because the sign of $(A_1)_j$ and coefficient λ_j can both change while the overall tensor is still fixed. Note that the coefficient λ_j can be positive or negative. According to the Fourier method for estimating b_1, mis-calculating the sign of $(A_1)_j$ also leads to sign of $b_1(j)$ recovered in the opposite manner. In other words, the recovered sign of the bias $b_1(j)$ is consistent with the recovered sign of $(A_1)_j$.

Recall we assume that the nonlinear activating function $\sigma(z)$ satisfies the property such that $\sigma(z) = 1 - \sigma(-z)$. Many popular activating functions such as step function, sigmoid function and tanh function satisfy this property. Given this property, the sign ambiguity in parameters A_1 and b_1 which leads to opposite sign in input z to the activating function $\sigma(\cdot)$ can be now compensated by the sign of a_2 and value of b_2, which is recovered through least squares.

Proof of Lemma D.1: From Lemma 4.1, we know that the exact cross-moment $\tilde{T} = \mathbb{E}[\tilde{y} \cdot \mathcal{S}_3(x)]$ has rank-one components as columns of matrix A_1; see Equation (4.16) for the tensor decomposition form. We apply a tensor decomposition method in NN-LIFT to estimate the columns of A_1. We

employ noisy tensor decomposition guarantees in Anandkumar et al. [22]. They show that when the perturbation tensor is small, the tensor power iteration initialized by the SVD-based Procedure 13 recovers the rank-1 components up to some small error. We also analyze the whitening step and combine it with this result leading to Lemma D.2.

Let us now characterize the perturbation matrix and tensor. By Lemma 4.1, the CP decomposition form is given by $\tilde{T} = \mathbb{E}[\tilde{y} \cdot \mathcal{S}_3(x)]$, and thus, the perturbation tensor is written as

$$E := \tilde{T} - \widehat{T} = \mathbb{E}[\tilde{y} \cdot \mathcal{S}_3(x)] - \frac{1}{n} \sum_{i \in [n]} \tilde{y}_i \cdot \mathcal{P}_3(x_i), \tag{D.4}$$

where $\widehat{T} = \frac{1}{n} \sum_{i \in [n]} \tilde{y}_i \cdot \mathcal{P}_3(x_i)$ is the empirical form used in NN-LIFT Algorithm 6. Notice that in the realizable setting, the neural network output $\tilde{y}$ is observed and thus, it is used in forming the empirical tensor. Similarly, the perturbation of second order moment $\tilde{M}_2 = \mathbb{E}[\tilde{y} \cdot \mathcal{S}_2(x)]$ is given by

$$E_2 := \tilde{M}_2 - \widehat{M}_2 = \mathbb{E}[\tilde{y} \cdot \mathcal{S}_2(x)] - \frac{1}{n} \sum_{i \in [n]} \tilde{y}_i \cdot \mathcal{P}_2(x_i). \tag{D.5}$$

In order to bound $\|E\|$, we matricize it to apply matrix Bernstein's inequality. We have the matricized version as

$$\tilde{E} := \mathbb{E}[\tilde{y} \cdot M_3(x)] - \frac{1}{n} \sum_{i \in [n]} \tilde{y}_i \cdot M_3(x_i) = \sum_{i \in [n]} \frac{1}{n} \Big(\mathbb{E}[\tilde{y} \cdot M_3(x)] - \tilde{y}_i \cdot M_3(x_i) \Big),$$

where $M_3(x) \in \mathbb{R}^{d \times d^2}$ is the matricization of $\mathcal{S}_3(x) \in \mathbb{R}^{d \times d \times d}$; see (1.1) for the definition of matricization. Now the norm of $\tilde{E}$ can be bounded by the matrix Bernstein's inequality. The norm of each (centered) random variable inside the summation is bounded as $\frac{\tilde{y}_{\max}}{n} \mathbb{E}[\|M_3(x)\|]$, where $\tilde{y}_{\max}$ is the bound on $|\tilde{y}|$. The variance term is also bounded as

$$\frac{1}{n^2} \Big\| \sum_{i \in [n]} \mathbb{E}\left[\tilde{y}_i^2 \cdot M_3(x_i) M_3^\top(x_i) \right] \Big\| \leq \frac{1}{n} \tilde{y}_{\max}^2 \mathbb{E}\left[\Big\| M_3(x) M_3^\top(x) \Big\| \right].$$

Applying matrix Bernstein's inequality, we have w.h.p.

$$\|E\| \leq \|\tilde{E}\| \leq \tilde{O}\left(\frac{\tilde{y}_{\max}}{\sqrt{n}}\sqrt{\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]}\right). \tag{D.6}$$

For the second order perturbation E_2, it is already a matrix, and by applying matrix Bernstein's inequality, we similarly argue that w.h.p.

$$\|E_2\| \leq \tilde{O}\left(\frac{\tilde{y}_{\max}}{\sqrt{n}}\sqrt{\mathbb{E}\left[\left\|\mathcal{S}_2(x)\mathcal{S}_2^\top(x)\right\|\right]}\right). \tag{D.7}$$

There is one more remaining piece to complete the proof of tensor decomposition part. The analysis in Anandkumar et al. [22] does not involve any whitening step, and thus, we need to adapt the perturbation analysis of Anandkumar et al. [22] to our additional whitening procedure. This is done in Lemma D.2. In the final recovery bound (D.17) in Lemma D.2, there are two terms; one involving $\|E\|$, and the other involving $\|E_2\|$. We first impose a bound on sample complexity such that the bound involving $\|E\|$ dominates the bound involving $\|E_2\|$ as follows. Considering the bounds on $\|E\|$ and $\|E_2\|$ in (D.6) and (D.7), and imposing the lower bound on the number of samples (third bound stated in the lemma) as

$$n \geq \tilde{O}\left(\tilde{y}_{\max}^2 \cdot \frac{\mathbb{E}\left[\left\|\mathcal{S}_2(x)\mathcal{S}_2^\top(x)\right\|\right]^{3/2}}{\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]^{1/2}} \cdot \frac{1}{\tilde{\lambda}_{\min}^2 \cdot s_{\min}^3(A_1)}\right),$$

leads to this goal. By doing this, we do not need to impose the bound on $\|E_2\|$ anymore, and applying the perturbation bound in (D.6) to the required bound on $\|E\|$ in Lemma D.2 leads to sample complexity bound (second bound stated in the lemma)

$$n \geq \tilde{O}\left(\tilde{y}_{\max}^2 \cdot \mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right] \cdot \left(\frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}\right)^3 \frac{1}{\lambda_{\min}^2 \cdot s_{\min}^6(A_1)} \cdot k\right).$$

Finally, applying the result of Lemma D.2, we have the column-wise error guarantees (up to permutation)

$$\|(A_1)_j - (\widehat{A}_1)_j\| \leq \tilde{O}\left(\frac{s_{\max}(A_1)}{\lambda_{\min}} \frac{\tilde{\lambda}_{\max}^2}{\sqrt{\tilde{\lambda}_{\min}}} \frac{\tilde{y}_{\max}}{\tilde{\lambda}_{\min}^{1.5} \cdot s_{\min}^3(A_1)} \frac{\sqrt{\mathbb{E}\left[\left\|M_3(x) M_3^\top(x)\right\|\right]}}{\sqrt{n}}\right) \leq \tilde{O}\left(\tilde{\epsilon}_1\right),$$

where in the first inequality we also substituted the bound on $\|E\|$ in (D.6), and the first bound on n stated in the lemma is used in the last inequality. □

D.2.1.1 Whitening analysis

The perturbation analysis of proposed tensor decomposition method in Algorithm 12 with the corresponding SVD-based initialization in Procedure 13 is provided in Anandkumar et al. [22]. But, they do not consider the effect of whitening proposed in Procedures 10 and 11. Thus, we need to adapt the perturbation analysis of Anandkumar et al. [22] when the whitening procedure is incorporated. We perform it in this section.

We first elaborate on the whitening step, and analyze how the proposed Procedure 10 works. We then analyze the inversion of whitening operator showing how the components in the whitened space are translated back to the original space as stated in Procedure 11. We finally provide the perturbation analysis of whitening step when estimations of moments are given.

Whitening procedure: Consider second order moment $\tilde{M}_2$ which is used to whiten third order tensor

$$\tilde{T} = \sum_{j \in [k]} \lambda_j \cdot (A_1)_j \otimes (A_1)_j \otimes (A_1)_j \tag{D.8}$$

in Procedure 10. It is constructed such that it has the same decomposition form as target tensor $\tilde{T}$, i.e., we have

$$\tilde{M}_2 = \sum_{j \in [k]} \tilde{\lambda}_j \cdot (A_1)_j \otimes (A_1)_j. \tag{D.9}$$

We propose two options for constructing $\tilde{M}_2$ in Procedure 10. First option is to use second order score function and construct $\tilde{M}_2 := \mathbb{E}\left[\tilde{y} \cdot \mathcal{P}_2(x)\right]$ for which we have

$$\tilde{M}_2 := \mathbb{E}\left[\tilde{y} \cdot \mathcal{P}_2(x)\right] = \sum_{j \in [k]} \tilde{\lambda}_j \cdot (A_1)_j \otimes (A_1)_j, \tag{D.10}$$

where

$$\tilde{\lambda}_j = \mathbb{E}\left[\sigma''(z_j)\right] \cdot a_2(j), \tag{D.11}$$

for vector $z := A_1^\top x + b_1$ as the input to the nonlinear operator $\sigma(\cdot)$. This is proved similar to Lemma 4.1. Second option leads to the same form for $\tilde{M}_2$ as (D.9) with coefficient modified as $\tilde{\lambda}_j = \lambda_j \cdot \langle (A_1)_j, \theta \rangle$.

Let matrix $W \in \mathbb{R}^{d \times k}$ denote the whitening matrix in the noiseless case, i.e., the whitening matrix W in Procedure 10 is constructed such that $W^\top \tilde{M}_2 W = I_k$. Applying whitening matrix W to the noiseless tensor $\tilde{T} = \sum_{j \in [k]} \lambda_j \cdot (A_1)_j \otimes (A_1)_j \otimes (A_1)_j$, we have

$$\tilde{T}(W, W, W) = \sum_{j \in [k]} \lambda_j \left(W^\top (A_1)_j\right)^{\otimes 3} = \sum_{j \in [k]} \frac{\lambda_j}{\tilde{\lambda}_j^{3/2}} \left(W^\top (A_1)_j \sqrt{\tilde{\lambda}_j}\right)^{\otimes 3} = \sum_{j \in [k]} \mu_j v_j^{\otimes 3}, \tag{D.12}$$

where we define

$$\mu_j := \frac{\lambda_j}{\tilde{\lambda}_j^{3/2}}, \quad v_j := W^\top (A_1)_j \sqrt{\tilde{\lambda}_j}, \quad j \in [k], \tag{D.13}$$

in the last equality. Let $V := [v_1 \ v_2 \ \cdots \ v_k] \in \mathbb{R}^{k \times k}$ denote the factor matrix for $\tilde{T}(W, W, W)$. We have

$$V := W^\top A_1 \operatorname{Diag}(\tilde{\lambda}^{1/2}), \tag{D.14}$$

and thus,

$$V V^\top = W^\top A_1 \operatorname{Diag}(\tilde{\lambda}) A_1^\top W = W^\top \tilde{M}_2 W = I_k.$$

Since V is a square matrix, it is also concluded that $V^\top V = I_k$, and therefore, tensor $\tilde{T}(W, W, W)$ is whitened such that the rank-1 components v_j's form an orthonormal basis. This discussion clarifies how the whitening procedure works.

Inversion of the whitening procedure: Let us also analyze the inversion procedure on how to transform v_j's to $(A_1)_j$'s. The main step is stated in Procedure 11. According to whitening Procedure 10, let $\tilde{M}_2 = U \operatorname{Diag}(\gamma) U^\top$, $U \in \mathbb{R}^{d \times k}$, $\gamma \in \mathbb{R}^k$, denote the rank-k SVD of $\tilde{M}_2$. Substituting whitening matrix $W := U \operatorname{Diag}(\gamma^{-1/2})$ in (D.14), and multiplying $U \operatorname{Diag}(\gamma^{1/2})$ from left, we have

$$U \operatorname{Diag}(\gamma^{1/2}) V = U U^\top A_1 \operatorname{Diag}(\tilde{\lambda}^{1/2}).$$

Since the column spans of $A_1 \in \mathbb{R}^{d \times k}$ and $U \in \mathbb{R}^{d \times k}$ are the same (given their relations to $\tilde{M}_2$), A_1 is a fixed point for the projection operator on the subspace spanned by the columns of U. This projector operator is $UU^\top$ (since columns of U form an orthonormal basis), and therefore, $UU^\top A_1 = A_1$. Applying this to the above equation, we have

$$A_1 = U \operatorname{Diag}(\gamma^{1/2}) V \operatorname{Diag}(\tilde{\lambda}^{-1/2}),$$

i.e.,

$$(A_1)_j = \frac{1}{\sqrt{\tilde{\lambda}_j}} U \operatorname{Diag}(\gamma^{1/2}) v_j, \quad j \in [k]. \tag{D.15}$$

The above discussions describe the details of whitening and unwhitening procedures. We now provide the guarantees of tensor decomposition given noisy versions of moments $\tilde{M}_2$ and $\tilde{T}$.

Lemma D.2. *Let $\hat{M}_2 = \tilde{M}_2 - E_2$ and $\widehat{T} = \tilde{T} - E$ respectively denote the noisy versions of*

$$\tilde{M}_2 = \sum_{j \in [k]} \tilde{\lambda}_j \cdot (A_1)_j \otimes (A_1)_j, \quad \tilde{T} = \sum_{j \in [k]} \lambda_j \cdot (A_1)_j \otimes (A_1)_j \otimes (A_1)_j. \tag{D.16}$$

Assume the second and third order perturbations satisfy the bounds

$$\|E_2\| \leq \tilde{O}\left(\lambda_{\min}^{1/3}\frac{\tilde{\lambda}_{\min}^{7/6}}{\sqrt{\tilde{\lambda}_{\max}}}s_{\min}^2(A_1)\frac{1}{k^{1/6}}\right),$$
$$\|E\| \leq \tilde{O}\left(\lambda_{\min}\left(\frac{\tilde{\lambda}_{\min}}{\tilde{\lambda}_{\max}}\right)^{1.5} s_{\min}^3(A_1)\frac{1}{\sqrt{k}}\right).$$

Then, the proposed tensor decomposition algorithm recovers estimations of rank-1 components $(A_1)_j$*'s satisfying error*

$$\|(A_1)_j - (\widehat{A}_1)_j\| \leq \tilde{O}\left(\frac{s_{\max}(A_1)}{\lambda_{\min}}\cdot\frac{\tilde{\lambda}_{\max}^2}{\sqrt{\tilde{\lambda}_{\min}}}\cdot\left[\frac{\|E_2\|^3}{\tilde{\lambda}_{\min}^{3.5}\cdot s_{\min}^6(A_1)}+\frac{\|E\|}{\tilde{\lambda}_{\min}^{1.5}\cdot s_{\min}^3(A_1)}\right]\right), \quad j\in[k]. \tag{D.17}$$

Proof: We do not have access to the true matrix $\tilde{M}_2$ and the true tensor $\tilde{T}$, and the perturbed versions $\hat{M}_2 = \tilde{M}_2 - E_2$ and $\widehat{T} = \tilde{T} - E$ are used in the whitening procedure. Here, $E_2 \in \mathbb{R}^{d\times d}$ denotes the perturbation matrix, and $E \in \mathbb{R}^{d\times d\times d}$ denotes the perturbation tensor. Similar to the noiseless case, let $\widehat{W} \in \mathbb{R}^{d\times k}$ denotes the whitening matrix constructed by Procedure 10 such that $\widehat{W}^\top \hat{M}_2 \widehat{W} = I_k$, and thus it orthogonalizes the noisy matrix $\hat{M}_2$. Applying the whitening matrix $\widehat{W}$ to the tensor $\widehat{T}$, we have

$$\begin{aligned}\widehat{T}(\widehat{W},\widehat{W},\widehat{W}) &= \tilde{T}(W,W,W) - \tilde{T}(W-\widehat{W},W-\widehat{W},W-\widehat{W}) - E(\widehat{W},\widehat{W},\widehat{W})\\ &= \sum_{j\in[k]}\mu_j v_j^{\otimes 3} - E_W,\end{aligned} \tag{D.18}$$

where we used Equation (D.12), and we defined

$$E_W := \tilde{T}(W-\widehat{W},W-\widehat{W},W-\widehat{W}) + E(\widehat{W},\widehat{W},\widehat{W}) \tag{D.19}$$

as the perturbation tensor after whitening. Note that the perturbation is from two sources; one is from the error in computing whitening matrix reflected in $W - \widehat{W}$, and the other is the error in tensor $\widehat{T}$ reflected in E.

We know that the rank-1 components v_j's form an orthonormal basis, and thus, we have a noisy orthogonal tensor decomposition problem in (D.18). We apply the result of Anandkumar et al. [22] where they show that if

$$\|E_W\| \leq \frac{\mu_{\min}\sqrt{\log k}}{\alpha_0\sqrt{k}},$$

for some constant $\alpha_0 > 1$, then the tensor power iteration (applied to the whitened tensor) recovers the tensor rank-1 components with bounded error (up to the permutation of columns)

$$\|v_j - \widehat{v}_j\| \leq \tilde{O}\left(\frac{\|E_W\|}{\mu_{\min}}\right). \tag{D.20}$$

We now relate the norm of E_W to the norm of original perturbations E and E_2. For the first term in (D.19), from Lemmata 4 and 5 of Song et al. [147], we have

$$\|\tilde{T}(W - \widehat{W}, W - \widehat{W}, W - \widehat{W})\| \leq \frac{64\|E_2\|^3}{\tilde{\lambda}_{\min}^{3.5} \cdot s_{\min}^6(A_1)}.$$

For the second term, by the sub-multiplicative property we have

$$\|E(\widehat{W}, \widehat{W}, \widehat{W})\| \leq \|E\| \cdot \|\widehat{W}\|^3 \leq 8\|E\| \cdot \|W\|^3 \leq \frac{8\|E\|}{s_{\min}^3(A_1)\tilde{\lambda}_{\min}^{3/2}}.$$

Here in the last inequality, we used

$$\|W\| = \frac{1}{\sqrt{s_k(\tilde{M}_2)}} \leq \frac{1}{s_{\min}(A_1)\sqrt{\tilde{\lambda}_{\min}}},$$

where $s_k(\tilde{M}_2)$ denotes the k-th largest singular value of $\tilde{M}_2$. Here, the equality is from the definition of W based on rank-k SVD of $\tilde{M}_2$ in Procedure 10, and the inequality is from $\tilde{M}_2 = A_1 \operatorname{Diag}(\tilde{\lambda}) A_1^\top$.

Substituting these bounds, we finally need the condition

$$\frac{64\|E_2\|^3}{\tilde{\lambda}_{\min}^{3.5} \cdot s_{\min}^6(A_1)} + \frac{8\|E\|}{\tilde{\lambda}_{\min}^{1.5} \cdot s_{\min}^3(A_1)} \leq \frac{\lambda_{\min}\sqrt{\log k}}{\alpha_0 \tilde{\lambda}_{\max}^{1.5}\sqrt{k}},$$

where we also substituted bound $\mu_{\min} \geq \lambda_{\min}/\tilde{\lambda}_{\max}^{1.5}$, given Equation (D.13). The bounds stated in the lemma ensures that each of the terms on the left hand side of the inequality are bounded by the right hand side. Thus, by the result of Anandkumar et al. [22], we have $\|v_j - \widehat{v}_j\| \leq \tilde{O}\left(\|E_W\|/\mu_{\min}\right)$. On the other hand, by the unwhitening relationship in (D.15), we have

$$\|(A_1)_j - (\widehat{A}_1)_j\| = \frac{1}{\sqrt{\tilde{\lambda}_j}} \|\operatorname{Diag}(\gamma^{1/2}) \cdot [v_j - \widehat{v}_j]\| \leq \sqrt{\frac{\gamma_{\max}}{\tilde{\lambda}_{\min}}} \cdot \|v_j - \widehat{v}_j\| \leq s_{\max}(A_1) \cdot \sqrt{\frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}} \cdot \|v_j - \widehat{v}_j\|. \tag{D.21}$$

where in the equality, we use the fact that orthonormal matrix U preserves the ℓ_2 norm, and the sub-multiplicative property is exploited in the first inequality. The last inequality is also from $\gamma_{\max} = s_{\max}(\tilde{M}_2) \leq s_{\max}^2(A_1) \cdot \tilde{\lambda}_{\max}$, which is from $\tilde{M}_2 = A_1 \operatorname{Diag}(\tilde{\lambda}) A_1^\top$. Incorporating the error bound on $\|v_j - \widehat{v}_j\|$ in (D.20), we have

$$\|(A_1)_j - (\widehat{A}_1)_j\| \leq \tilde{O}\left(s_{\max}(A_1) \cdot \sqrt{\frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}} \cdot \frac{\|E_W\|}{\mu_{\min}}\right) \leq \tilde{O}\left(\frac{s_{\max}(A_1)}{\lambda_{\min}} \cdot \frac{\tilde{\lambda}_{\max}^2}{\sqrt{\tilde{\lambda}_{\min}}} \cdot \|E_W\|\right),$$

where we used the bound $\mu_{\min} \geq \lambda_{\min}/\tilde{\lambda}_{\max}^{1.5}$ in the last step. □

D.2.2 Fourier analysis guarantees

The analysis of Fourier method for estimating parameter b_1 includes the following two lemmas. In the first lemma, we argue the mean of random variable v introduced in Algorithm 6 in the realizable setting. This clarifies why the phase of v is related to unknown parameter b_1. In the second lemma, we argue the concentration of v around its mean leading to the sample complexity result. Note that v is denoted by $\tilde{v}$ in the realizable setting.

The Fourier method can be also used to estimate the weight vector a_2 since it appears in the magnitude of complex number v. In this section, we also provide the analysis of estimating a_2 with Fourier method which can be used as an alternative, while we primarily estimate a_2 by the ridge regression analyzed in Appendix D.2.3.

Lemma 4.6 (Restated)**.** *Let*

$$\tilde{v} := \frac{1}{n}\sum_{i\in[n]} \frac{\tilde{y}_i}{p(x_i)} e^{-j\langle \omega_i, x_i\rangle}. \tag{D.22}$$

Notice this is a realizable of v in Procedure 7 where the output corresponds to a neural network $\tilde{y}$. If ω_i's are uniformly i.i.d. drawn from set Ω_l, then $\tilde{v}$ has mean (which is computed over x, $\tilde{y}$ and ω)

$$\mathbb{E}[\tilde{v}] = \frac{1}{|\Omega_l|}\Sigma\left(\frac{1}{2}\right) a_2(l) e^{j\pi b_1(l)}, \tag{D.23}$$

where $|\Omega_l|$ denotes the surface area of $d-1$ dimensional manifold Ω_l, and $\Sigma(\cdot)$ denotes the Fourier transform of $\sigma(\cdot)$.

Proof: Let $\tilde{F}(\omega)$ denote the Fourier transform of label function $\tilde{f}(x) := \mathbb{E}[\tilde{y}|x] = \langle a_2, \sigma(A_1^\top x + b_1)\rangle$ which is [121]

$$\tilde{F}(\omega) = \sum_{j\in[k]} \frac{a_2(j)}{|A_1(d,j)|}\Sigma\left(\frac{\omega_d}{A_1(d,j)}\right) e^{j2\pi b_1(j)\frac{\omega_d}{A_1(d,j)}} \delta\left(\omega_- - \frac{\omega_d}{A_1(d,j)} A_1(\backslash d, j)\right), \tag{D.24}$$

where $\Sigma(\cdot)$ is the Fourier transform of $\sigma(\cdot)$, $u_-^\top = [u_1, u_2, \ldots, u_{d-1}]$ is vector $u^\top$ with the last entry removed, $A_1(\backslash d, j) \in \mathbb{R}^{d-1}$ is the j-th column of matrix A_1 with the d-th (last) entry removed, and finally $\delta(u) = \delta(u_1)\delta(u_2)\cdots\delta(u_d)$.

Let $p(\omega)$ denote the probability density function of frequency ω. We have

$$\begin{aligned}
\mathbb{E}[\tilde{v}] &= \mathbb{E}_{x,\tilde{y},\omega}\left[\frac{\tilde{y}}{p(x)} e^{-j\langle\omega,x\rangle}\right] \\
&= \mathbb{E}_{x,\omega}\left[\mathbb{E}_{\tilde{y}|\{x,\omega\}}\left[\frac{\tilde{y}}{p(x)} e^{-j\langle\omega,x\rangle}\middle| x,\omega\right]\right] \\
&= \mathbb{E}_{x,\omega}\left[\frac{\tilde{f}(x)}{p(x)} e^{-j\langle\omega,x\rangle}\right] \\
&= \int_{\Omega_l}\int \tilde{f}(x) e^{-j\langle\omega,x\rangle} p(\omega) dx d\omega \\
&= \int_{\Omega_l} \tilde{F}(\omega) p(\omega) d\omega,
\end{aligned}$$

where the second equality uses the law of total expectation, the third equality exploits the label-generating function definition $\tilde{f}(x) := \mathbb{E}[\tilde{y}|x]$, and the final equality is from the definition of Fourier transform. The variable $\omega \in \mathbb{R}^d$ is drawn from a $d-1$ dimensional manifold $\Omega_l \subset \mathbb{R}^d$. In order to compute the above integral, we define d dimensional set

$$\Omega_{l;\nu} := \left\{ \omega \in \mathbb{R}^d : \frac{1}{2} - \frac{\nu}{2} \leq \|\omega\| \leq \frac{1}{2} + \frac{\nu}{2}, \left|\langle \omega, (\widehat{A}_1)_l \rangle\right| \geq \frac{1 - \tilde{\epsilon}_1^2/2}{2} \right\},$$

for which $\Omega_l = \lim_{\nu \to 0^+} \Omega_{l;\nu}$. Assuming ω's are uniformly drawn from $\Omega_{l;\nu}$, we have

$$\begin{aligned}
\mathbb{E}[\tilde{v}] &= \lim_{\nu \to 0^+} \int_{\Omega_{l;\nu}} \tilde{F}(\omega) p(\omega) d\omega \\
&= \lim_{\nu \to 0^+} \frac{1}{|\Omega_{l;\nu}|} \int_{-\infty}^{+\infty} \tilde{F}(\omega) 1_{\Omega_{l;\nu}}(\omega) d\omega.
\end{aligned}$$

The second equality is from uniform draws of ω from set $\Omega_{l;\nu}$ such that $p(\omega) = \frac{1}{|\Omega_{l;\nu}|} 1_{\Omega_{l;\nu}}(\omega)$, where $1_S(\cdot)$ denotes the indicator function for set S. Here, $|\Omega_{l;\nu}|$ denotes the volume of d dimensional subspace $\Omega_{l;\nu}$, for which in the limit $\nu \to 0^+$, we have $|\Omega_{l;\nu}| = \nu \cdot |\Omega_l|$, where $|\Omega_l|$ denotes the surface area of $d-1$ dimensional manifold Ω_l.

For small enough $\tilde{\epsilon}_1$ in the definition of $\Omega_{l;\nu}$, only the delta function for $j = l$ in the expansion of $\tilde{F}(\omega)$ in (D.24) is survived from the above integral, and thus,

$$\mathbb{E}[\tilde{v}] = \lim_{\nu \to 0^+} \frac{1}{|\Omega_{l;\nu}|} \int_{-\infty}^{+\infty} \frac{a_2(l)}{|A_1(d,l)|} \Sigma\left(\frac{\omega_d}{A_1(d,l)}\right) e^{j2\pi b_1(l) \frac{\omega_d}{A_1(d,l)}} \delta\left(\omega_- - \frac{\omega_d}{A_1(d,l)} A_1(\backslash d, l)\right) 1_{\Omega_{l;\nu}}(\omega) d\omega.$$

In order to simplify the notations, in the rest of the proof we denote l-th column of matrix A_1 by vector α, i.e., $\alpha := (A_1)_l$. Thus, the goal is to compute the integral

$$I := \int_{-\infty}^{+\infty} \frac{1}{|\alpha_d|} \Sigma\left(\frac{\omega_d}{\alpha_d}\right) e^{j2\pi b_1(l) \frac{\omega_d}{\alpha_d}} \delta\left(\omega_- - \frac{\omega_d}{\alpha_d} \alpha_-\right) 1_{\Omega_{l;\nu}}(\omega) d\omega,$$

and note that $\mathbb{E}[\tilde{v}] = a_2(l) \cdot \lim_{\nu \to 0^+} \frac{I}{|\Omega_{l;\nu}|}$. The rest of the proof is about computing the above integral. The integral involves delta functions where the final value is expected to be computed at a single point specified by the intersection of line $\omega_- = \frac{\omega_d}{\alpha_d} \alpha_-$, and sphere $\|\omega\| = \frac{1}{2}$ (when we consider the limit $\nu \to 0^+$). This is based on the following integration property of delta functions

such that for function $g(\cdot) : \mathbb{R} \to \mathbb{R}$,

$$\int_{-\infty}^{+\infty} g(t)\delta(t)dt = g(0). \tag{D.25}$$

We first expand the delta function as follows.

$$\begin{aligned} I &= \int_{-\infty}^{+\infty} \frac{1}{|\alpha_d|}\Sigma\left(\frac{\omega_d}{\alpha_d}\right) e^{j2\pi b_1(l)\frac{\omega_d}{\alpha_d}}\delta\left(\omega_1 - \frac{\alpha_1}{\alpha_d}\omega_d\right)\cdots\delta\left(\omega_{d-1} - \frac{\alpha_{d-1}}{\alpha_d}\omega_d\right) 1_{\Omega_{l;\nu}}(\omega)d\omega, \\ &= \int\cdots\int_{-\infty}^{+\infty} \Sigma\left(\frac{\omega_d}{\alpha_d}\right) e^{j2\pi b_1(l)\frac{\omega_d}{\alpha_d}}\delta\left(\omega_1 - \frac{\alpha_1}{\alpha_d}\omega_d\right)\cdots\delta\left(\omega_{d-2} - \frac{\alpha_{d-2}}{\alpha_d}\omega_d\right) \\ &\qquad\qquad 1_{\Omega_{l;\nu}}(\omega)\cdot\delta\left(\alpha_d\omega_{d-1} - \alpha_{d-1}\omega_d\right)d\omega_1\cdots\omega_d, \end{aligned}$$

where we used the property $\frac{1}{|\beta|}\delta(t) = \delta(\beta t)$ in the second equality. Introducing new variable z, and applying the change of variable $\omega_d = \frac{1}{\alpha_{d-1}}(\alpha_d\omega_{d-1} - z)$, we have

$$\begin{aligned} I &= \int\cdots\int_{-\infty}^{+\infty} \Sigma\left(\frac{\omega_d}{\alpha_d}\right) e^{j2\pi b_1(l)\frac{\omega_d}{\alpha_d}}\delta\left(\omega_1 - \frac{\alpha_1}{\alpha_d}\omega_d\right)\cdots\delta\left(\omega_{d-2} - \frac{\alpha_{d-2}}{\alpha_d}\omega_d\right) \\ &\qquad\qquad 1_{\Omega_{l;\nu}}(\omega)\cdot\delta(z)d\omega_1\cdots d\omega_{d-1}\frac{dz}{\alpha_{d-1}}, \\ &= \int\cdots\int_{-\infty}^{+\infty} \frac{1}{\alpha_{d-1}}\Sigma\left(\frac{\omega_{d-1}}{\alpha_{d-1}}\right) e^{j2\pi b_1(l)\frac{\omega_{d-1}}{\alpha_{d-1}}}\delta\left(\omega_1 - \frac{\alpha_1}{\alpha_{d-1}}\omega_{d-1}\right)\cdots\delta\left(\omega_{d-2} - \frac{\alpha_{d-2}}{\alpha_{d-1}}\omega_{d-1}\right) \\ &\qquad\qquad 1_{\Omega_{l;\nu}}\left(\left[\omega_1, \omega_2, \ldots, \omega_{d-1}, \frac{\alpha_d}{\alpha_{d-1}}\omega_{d-1}\right]\right)d\omega_1\cdots d\omega_{d-1}. \end{aligned}$$

For the sake of simplifying the mathematical notations, we did not substitute all the ω_d's with z in the first equality, but note that all ω_d's are implicitly a function of z which is finally considered in the second equality where the delta integration property in (D.25) is applied to variable z (note that $z = 0$ is the same as $\frac{\omega_d}{\alpha_d} = \frac{\omega_{d-1}}{\alpha_{d-1}}$). Repeating the above process several times, we finally have

$$I = \int_{-\infty}^{+\infty} \frac{1}{\alpha_1}\Sigma\left(\frac{\omega_1}{\alpha_1}\right) e^{j2\pi b_1(l)\frac{\omega_1}{\alpha_1}}\cdot 1_{\Omega_{l;\nu}}\left(\left[\omega_1, \frac{\alpha_2}{\alpha_1}\omega_1, \ldots, \frac{\alpha_{d-1}}{\alpha_1}\omega_1, \frac{\alpha_d}{\alpha_1}\omega_1\right]\right)d\omega_1.$$

There is a line constraint as $\frac{\omega_1}{\alpha_1} = \frac{\omega_2}{\alpha_2} = \cdots = \frac{\omega_d}{\alpha_d}$ in the argument of indicator function. This implies that $\|\omega\| = \frac{\|\alpha\|}{\alpha_1}\omega_1 = \frac{\omega_1}{\alpha_1}$, where we used $\|\alpha\| = \|(A_1)_l\| = 1$. Incorporating this in the norm bound

imposed by the definition of $\Omega_{l;\nu}$, we have $\frac{\alpha_1}{2}(1-\nu) \leq \omega_1 \leq \frac{\alpha_1}{2}(1+\nu)$, and hence,

$$I = \int_{\frac{\alpha_1}{2}(1-\nu)}^{\frac{\alpha_1}{2}(1+\nu)} \frac{1}{\alpha_1} \Sigma\left(\frac{\omega_1}{\alpha_1}\right) e^{j2\pi b_1(l)\frac{\omega_1}{\alpha_1}} d\omega_1.$$

We know $\mathbb{E}[\tilde{v}] = a_2(l) \cdot \lim_{\nu \to 0^+} \frac{I}{|\Omega_{l;\nu}|}$, and thus,

$$\mathbb{E}[\tilde{v}] = a_2(l) \cdot \frac{1}{\nu \cdot |\Omega_l|} \cdot \alpha_1 \nu \frac{1}{\alpha_1} \Sigma\left(\frac{1}{2}\right) e^{j2\pi b_1(l)\frac{1}{2}} = \frac{1}{|\Omega_l|} a_2(l) \Sigma\left(\frac{1}{2}\right) e^{j\pi b_1(l)},$$

where in the first step we use $|\Omega_{l;\nu}| = \nu \cdot |\Omega_l|$, and write the integral I in the limit $\nu \to 0^+$. This finishes the proof. □

In the following lemma, we argue the concentration of v around its mean which leads to the sample complexity bound for estimating the parameter b_1 (and also a_2) within the desired error.

Lemma D.3. *If the sample complexity*

$$n \geq O\left(\frac{\tilde{\zeta}_{\tilde{f}}}{\psi \tilde{\epsilon}_2^2} \log \frac{k}{\delta}\right) \tag{D.26}$$

holds for small enough $\tilde{\epsilon}_2 \leq \tilde{\zeta}_{\tilde{f}}$, then the estimates $\widehat{a}_2(l) = \frac{|\Omega_l|}{|\Sigma(1/2)|}|\tilde{v}|$, and $\widehat{b}_1(l) = \frac{1}{\pi}(\angle\tilde{v} - \angle\Sigma(1/2))$ for $l \in [k]$, in NN-LIFT Algorithm 6 (see the definition of $\tilde{v}$ in (D.22)*) satisfy with probability at least $1-\delta$,*

$$|a_2(l) - \widehat{a}_2(l)| \leq \frac{|\Omega_l|}{|\Sigma(1/2)|} O(\tilde{\epsilon}_2), \quad |b_1(l) - \widehat{b}_1(l)| \leq \frac{|\Omega_l|}{\pi |\Sigma(1/2)||a_2(l)|} O(\tilde{\epsilon}_2).$$

Proof: The result is proved by arguing the concentration of variable $\tilde{v}$ in (D.22) around its mean characterized in (D.23). We use the Bernstein's inequality to do this. Let $\tilde{v} := \sum_{i\in[n]} \tilde{v}_i$ where $\tilde{v}_i = \frac{1}{n}\frac{\tilde{y}_i}{p(x_i)} e^{-j\langle \omega_i, x_i\rangle}$. By the lower bound $p(x) \geq \psi$ assumed in Theorem 4.3 and labels $\tilde{y}_i$'s being bounded, the magnitude of centered $\tilde{v}_i$'s $(\tilde{v}_i - \mathbb{E}[\tilde{v}_i])$ are bounded by $O(\frac{1}{\psi n})$. The variance term is also bounded as

$$\sigma^2 = \Big| \sum_{i\in[n]} \mathbb{E}\left[(\tilde{v}_i - \mathbb{E}[\tilde{v}_i])(\overline{\tilde{v}_i - \mathbb{E}[\tilde{v}_i]})\right] \Big|,$$

where $\overline{u}$ denotes the complex conjugate of complex number u. This is bounded as

$$\sigma^2 \leq \sum_{i\in[n]} \mathbb{E}\left[\tilde{v}_i\overline{\tilde{v}_i}\right] = \frac{1}{n^2}\sum_{i\in[n]} \mathbb{E}\left[\frac{\tilde{y}_i^2}{p(x_i)^2}\right]$$

Since output $\tilde{y}$ is a binary label ($\tilde{y} \in \{0,1\}$), we have $\mathbb{E}[\tilde{y}^2|x] = \mathbb{E}[\tilde{y}|x] = \tilde{f}(x)$, and thus,

$$\mathbb{E}\left[\frac{\tilde{y}^2}{p(x)^2}\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{\tilde{y}^2}{p(x)^2}|x\right]\right] = \mathbb{E}\left[\frac{\tilde{f}(x)}{p(x)^2}\right] \leq \frac{1}{\psi}\int_{\mathbb{R}^d} \tilde{f}(x)dx = \frac{\tilde{\zeta}_{\tilde{f}}}{\psi},$$

where the inequality uses the bound $p(x) \geq \psi$ and the last equality is from definition of $\tilde{\zeta}_{\tilde{f}}$. This provides us the bound on variance as

$$\sigma^2 \leq \frac{\tilde{\zeta}_{\tilde{f}}}{\psi n}.$$

Applying Bernstein's inequality concludes the concentration bound such that with probability at least $1-\delta$, we have

$$|\tilde{v} - \mathbb{E}[\tilde{v}]| \leq O\left(\frac{1}{\psi n}\log\frac{1}{\delta} + \sqrt{\frac{\tilde{\zeta}_{\tilde{f}}}{\psi n}\log\frac{1}{\delta}}\right) \leq O(\tilde{\epsilon}_2),$$

where the last inequality is from sample complexity bound. This implies that $||\tilde{v}| - |\mathbb{E}[\tilde{v}]|| \leq O(\tilde{\epsilon}_2)$. Substituting $|\mathbb{E}[\tilde{v}]|$ from (D.23) and considering estimate $\widehat{a}_2(l) = \frac{|\Omega_l|}{|\Sigma(1/2)|}|\tilde{v}|$, we have

$$|\widehat{a}_2(l) - a_2(l)| \leq \frac{|\Omega_l|}{|\Sigma(1/2)|}O(\tilde{\epsilon}_2),$$

which finishes the first part of the proof. For the phase, we have $\phi := \angle\tilde{v} - \angle\mathbb{E}[\tilde{v}] = \pi(\widehat{b}_1(l) - b_1(l))$. On the other hand, for small enough error $\tilde{\epsilon}_2$ (and thus small ϕ), we have the approximation $\phi \sim \tan(\phi) \sim \frac{|\tilde{v}-\mathbb{E}[\tilde{v}]|}{|\mathbb{E}[\tilde{v}]|}$ (note that this is actually an upper bound such that $\phi \leq \tan(\phi)$). Thus,

$$|\widehat{b}_1(l) - b_1(l)| \leq \frac{1}{\pi|\mathbb{E}[\tilde{v}]|}O(\tilde{\epsilon}_2) \leq \frac{|\Omega_l|}{\pi|\Sigma(1/2)||a_2(l)|}O(\tilde{\epsilon}_2).$$

This finishes the proof of second bound. □

D.2.3 Ridge regression analysis and guarantees

Let $h := \sigma(A_1^\top x + b_1)$ denote the neuron or hidden layer variable. With slightly abuse of notation, in the rest of analysis in this section, we append variable h by the dummy variable 1 to represent the bias, and thus, $h \in \mathbb{R}^{k+1}$. We write the output as $\tilde{y} = h^\top \beta + \eta$, where

$$\beta := [a_2, b_2] \in \mathbb{R}^{k+1}.$$

Given the estimated parameters of first layer denoted by $\widehat{A}_1$ and $\widehat{b}_1$, the neurons are estimated as $\widehat{h} := \sigma(\widehat{A}_1^\top x + \widehat{b}_1)$. In addition, the dummy variable 1 is also appended, and thus, $\widehat{h} \in \mathbb{R}^{k+1}$. Because of this estimated encoding of neurons, we expand the output $\tilde{y}$ as

$$\tilde{y} = \widehat{h}^\top \beta + \underbrace{(h^\top - \widehat{h}^\top)\beta}_{\text{bias (approximation): } b(\widehat{h})} + \underbrace{\eta}_{\text{noise}} = \tilde{f}(\widehat{h}) + \eta, \tag{D.27}$$

where $\tilde{f}(\widehat{h}) := \mathbb{E}[\tilde{y}|\widehat{h}] = \widehat{h}^\top \beta + b(\widehat{h})$. Here, we have a noisy linear model with additional bias (approximation). Let $\hat{\beta}_\lambda$ denote the ridge regression estimator for some regularization parameter $\lambda \geq 0$, which is defined as the minimizer of the regularized empirical mean squared error, i.e.,

$$\hat{\beta}_\lambda := \arg\min_\beta \frac{1}{n} \sum_{i \in [n]} \Big(\langle \beta, \widehat{h}_i \rangle - \tilde{y}_i \Big)^2 + \lambda \|\beta\|^2.$$

We know this estimator is given by (when $\hat{\Sigma}_{\widehat{h}} + \lambda I \succ 0$)

$$\hat{\beta}_\lambda = \Big(\hat{\Sigma}_{\widehat{h}} + \lambda I \Big)^{-1} \cdot \hat{\mathbb{E}}(\widehat{h}\tilde{y}),$$

where $\hat{\Sigma}_{\widehat{h}} := \frac{1}{n} \sum_{i \in [n]} \widehat{h}_i \widehat{h}_i^\top$ is the empirical covariance of $\widehat{h}$, and $\hat{\mathbb{E}}$ denotes the empirical mean operator. The analysis of ridge regression leads to the following expected prediction error (risk) bound on the estimation of the output.

Lemma D.4 (Expected prediction error of ridge regression)**.** *Suppose the parameter recovery results in Lemmata D.1 and D.3 on A_1 and b_1 hold. In addition, assume the nonlinear activating function $\sigma(\cdot)$ satisfies the Lipschitz property such that $|\sigma(u) - \sigma(u')| \leq L \cdot |u - u'|$, for $u, u' \in \mathbb{R}$. The*

following noise, approximation and statistical leverage conditions also hold. Then, by choosing the optimal $\lambda > 0$ in the λ-regularized ridge regression (which estimates the parameters $\widehat{a}_2$ and $\widehat{b}_2$), the estimated output as $\widehat{f}(x) = \widehat{a}_2^\top \sigma(\widehat{A}_1^\top x + \widehat{b}_1) + \widehat{b}_2$ satisfies the risk bound

$$\mathbb{E}[|\widehat{f}(x) - \tilde{f}(x)|^2] \leq O\left(\frac{k\|\beta\|^2}{n}\right) + O\left(\sqrt{\frac{2k\|\beta\|^2}{n}\left(\mathbb{E}[b(\widehat{h})^2] + \sigma_{noise}^2/2\right)}\right) + \mathbb{E}[b(\widehat{h})^2],$$

where

$$\mathbb{E}[b(\widehat{h})^2] \leq \left[r + \frac{|\Omega_l|}{\pi|\Sigma(1/2)||a_2(l)|}\right]^2 \|\beta\|^2 L^2 k O(\tilde{\epsilon}^2).$$

Proof: Since $\widehat{h} := \sigma(\widehat{A}_1^\top x + \widehat{b}_1)$, we equivalently argue the bound on $\mathbb{E}[(\widehat{h}^\top \hat{\beta}_\lambda - \tilde{f}(\widehat{h}))^2]$, where $\widehat{f}(x) = \widehat{f}(\widehat{h}) = \widehat{h}^\top \hat{\beta}_\lambda$. From standard results in the study of inverse problems, we know (see Proposition 5 in Hsu et al. [96])

$$\mathbb{E}[(\widehat{h}^\top \hat{\beta}_\lambda - \tilde{f}(\widehat{h}))^2] = \mathbb{E}[(\widehat{h}^\top \beta - \tilde{f}(\widehat{h}))^2] + \|\hat{\beta}_\lambda - \beta\|_{\Sigma_{\widehat{h}}}^2.$$

Here, for positive definite matrix $\Sigma \succ 0$, the vector norm $\|\cdot\|_\Sigma$ is defined as $\|v\|_\Sigma := \sqrt{v^\top \Sigma v}$. For the first term, by the definition of $\tilde{f}(\widehat{h})$ as $\tilde{f}(\widehat{h}) := \mathbb{E}[\tilde{y}|\widehat{h}] = \widehat{h}^\top \beta + b(\widehat{h})$, we have

$$\mathbb{E}[(\widehat{h}^\top \beta - \tilde{f}(\widehat{h}))^2] = \mathbb{E}[b(\widehat{h})^2].$$

Lemma D.5 bounds $\mathbb{E}[b(\widehat{h})^2]$ and bounding $\|\hat{\beta}_\lambda - \beta\|_{\Sigma_{\widehat{h}}}^2$ is argued in Lemma D.6 and Remark 30. Combining these bounds finishes the proof. □

In order to have final risk bounded as $\mathbb{E}[|\widehat{f}(x) - \tilde{f}(x)|^2] \leq \tilde{O}(\epsilon^2)$, for some $\epsilon > 0$, the above lemma imposes sample complexity as (some of other parameters considered in (D.3), (D.26) are not repeated here)

$$n \geq \tilde{O}\left(L\frac{k\|\beta\|^2}{\epsilon^2}(1 + \sigma_{\text{noise}}^2)\right). \tag{D.28}$$

Lemma D.5 (Bounded approximation)**.** *Suppose the parameter recovery results in Lemmata D.1 and D.3 on A_1 and b_1 hold. In addition, assume the nonlinear activating function $\sigma(\cdot)$ satisfies the Lipschitz*

property such that $|\sigma(u) - \sigma(u')| \leq L \cdot |u - u'|$*, for* $u, u' \in \mathbb{R}$*. Then, the approximation term is bounded as*

$$\mathbb{E}[b(\widehat{h})^2] \leq \left[r + \frac{|\Omega_l|}{\pi |\Sigma(1/2)||a_2(l)|} \right]^2 \|\beta\|^2 L^2 k O(\tilde{\epsilon}^2).$$

Proof: We have

$$\mathbb{E}[b(\widehat{h})^2] = \mathbb{E}[\langle h - \widehat{h}, \beta \rangle^2] \leq \|\beta\|^2 \cdot \mathbb{E}[\|h - \widehat{h}\|^2]. \tag{D.29}$$

Define $\tilde{\epsilon} := \max\{\tilde{\epsilon}_1, \tilde{\epsilon}_2\}$, where $\tilde{\epsilon}_1$ and $\tilde{\epsilon}_2$ are the corresponding bounds in Lemmata D.1 and D.3, respectively. Using the Lipschitz property of nonlinear function $\sigma(\cdot)$, we have

$$\begin{aligned}
|h_l - \widehat{h}_l| &= |\sigma(\langle (A_1)_l, x \rangle + b_1(l)) - \sigma(\langle (\widehat{A}_1)_l, x \rangle + \widehat{b}_1(l))| \\
&\leq L \cdot \left[|\langle (A_1)_l - (\widehat{A}_1)_l, x \rangle| + |b_1(l) - \widehat{b}_1(l)| \right] \\
&\leq L \cdot \left[r O(\tilde{\epsilon}) + \frac{|\Omega_l|}{\pi |\Sigma(1/2)||a_2(l)|} O(\tilde{\epsilon}) \right],
\end{aligned}$$

where in the second inequality, we use the bounds in Lemmata D.1 and D.3, and bounded x such that $\|x\| \leq r$. Applying this to (D.29) concludes the proof. □

We now assume the following additional conditions to bound $\|\hat{\beta}_\lambda - \beta\|^2_{\Sigma_{\widehat{h}}}$. The following discussions are along the results of Hsu et al. [96].

We define the effective dimensions of the covariate $\widehat{h}$ as

$$k_{p,\lambda} := \sum_{j \in [k]} \left(\frac{\lambda_j}{\lambda_j + \lambda} \right)^p, \quad p \in \{1, 2\},$$

where λ_j's denote the (positive) eigenvalues of $\Sigma_{\widehat{h}}$, and λ is the regularization parameter of ridge regression.

- Subgaussian noise: there exists a finite $\sigma_{\text{noise}} \geq 0$ such that, almost surely,

$$\mathbb{E}_\eta[\exp(\alpha \eta) | \widehat{h}] \leq \exp(\alpha^2 \sigma^2_{\text{noise}} / 2), \quad \forall \alpha \in \mathbb{R},$$

where η denotes the noise in the output $\tilde{y}$.

- Bounded statistical leverage: there exists a finite $\rho_\lambda \geq 1$ such that, almost surely,

$$\frac{\sqrt{k}}{\sqrt{(\inf\{\lambda_j\} + \lambda)k_{1,\lambda}}} \leq \rho_\lambda.$$

- Bounded approximation error at λ: there exists a finite $B_{\text{bias},\lambda} \geq 0$ such that, almost surely,

$$\rho_\lambda \left(B_{\max} + \sqrt{k}\|\beta\| \right) \leq B_{\text{bias},\lambda},$$

  where $|b(\widehat{h})| \leq B_{\max}$. Note that the approximation term $b(\widehat{h})$ is bounded in Lemma D.5. The parameter $B_{\text{bias},\lambda}$ only contributes to the lower order terms in the analysis of ridge regression.

Lemma D.6 (Bounding excess mean squared error: Theorem 2 of Hsu et al. [96])**.** *Fix some $\lambda \geq 0$, and suppose the above noise, approximation and statistical leverage conditions hold, and in addition,*

$$n \geq \tilde{O}(\rho_\lambda^2 k_{1,\lambda}). \tag{D.30}$$

Then, we have

$$\|\hat{\beta}_\lambda - \beta\|_{\Sigma_{\widehat{h}}}^2 \leq \tilde{O}\left(\frac{k}{\lambda n} \left(\mathbb{E}[b(\widehat{h})^2] + \sigma_{noise}^2/2 \right) + \frac{\lambda\|\beta\|^2}{2} \left(1 + \frac{k/\lambda + 1}{n} \right) \right) + o(1/n),$$

where $\mathbb{E}[b(\widehat{h})^2]$ is bounded in Lemma D.5.

In the above lemma, we also used the discussions in Remarks 12 and 15 of Hsu et al. [96] which include comments on the simplification of the general result.

Remark 30 (Optimal λ). In addition, along the discussion in Remark 15 of Hsu et al. [96], by choosing the optimal $\lambda > 0$ that minimizes the bound in the above lemma, we have

$$\|\hat{\beta}_\lambda - \beta\|_{\Sigma_{\widehat{h}}}^2 \leq O\left(\frac{k\|\beta\|^2}{n} \right) + O\left(\sqrt{\frac{2k\|\beta\|^2}{n} \left(\mathbb{E}[b(\widehat{h})^2] + \sigma_{\text{noise}}^2/2 \right)} \right).$$

D.3 Proof of Theorem 4.5

Before we provide the proof, we first state the details of bound on C_f. We require

$$C_f \leq \min \left\{ \tilde{O}\left(\frac{1}{r}\left(\frac{1}{\sqrt{k}}+\delta_1\right)^{-1} \frac{1}{\sqrt{\mathbb{E}[\|\mathcal{S}_3(x)\|^2]}} \cdot \frac{\tilde{\lambda}_{\min}^2}{\tilde{\lambda}_{\max}^2} \cdot \lambda_{\min} \cdot \frac{s_{\min}^3(A_1)}{s_{\max}(A_1)} \cdot \tilde{\epsilon}_1 \right), \right. \tag{D.31}$$

$$\tilde{O}\left(\frac{1}{r}\left(\frac{1}{\sqrt{k}}+\delta_1\right)^{-1} \frac{1}{\sqrt{\mathbb{E}[\|\mathcal{S}_3(x)\|^2]}} \cdot \lambda_{\min} \left(\frac{\tilde{\lambda}_{\min}}{\tilde{\lambda}_{\max}}\right)^{1.5} s_{\min}^3(A_1) \cdot \frac{1}{\sqrt{k}} \right),$$

$$\left. O\left(\frac{1}{r}\left(\frac{1}{\sqrt{k}}+\delta_1\right)^{-1} \frac{\mathbb{E}[\|\mathcal{S}_3(x)\|^2]^{1/4}}{\mathbb{E}[\|\mathcal{S}_2(x)\|^2]^{3/4}} \cdot \tilde{\lambda}_{\min} \cdot s_{\min}^{1.5}(A_1) \right) \right\}.$$

Proof of Theorem 4.5: We first argue that the perturbation involves both estimation and approximation parts. **Perturbation decomposition into approximation and estimation parts:** Similar to the estimation part analysis, we need to ensure the perturbation from exact means is small enough to apply the analysis of Lemmas D.1 and D.3. Here, in addition to the empirical estimation of quantities (estimation error), the approximation error also contributes to the perturbation. This is because there is no realizable setting here, and the observations are from an arbitrary function $f(x)$. We address this for both the tensor decomposition and the Fourier parts as follows.

Recall that we use notation $\tilde{f}(x)$ (and $\tilde{y}$) to denote the output of a neural network. For arbitrary function $f(x)$, we refer to the neural network satisfying the approximation error provided in Theorem 4.4 by $\tilde{y}_f$. The ultimate goal of our analysis is to show that NN-LIFT recovers the parameters of this specific neural network with small error. More precisely, note that these are a class of neural networks satisfying the approximation bound in Theorem 4.5, and it suffices to say that the output of the algorithm is close enough to one of them.

Tensor decomposition: There are two perturbation sources in the tensor analysis. One is from the approximation part and the other is from the estimation part. By Lemma 4.1, the CP decom-

position form is given by $\tilde{T}_f = \mathbb{E}[\tilde{y}_f \cdot \mathcal{S}_3(x)]$, and thus, the perturbation tensor is written as

$$E := \tilde{T}_f - \widehat{T} = \mathbb{E}[\tilde{y}_f \cdot \mathcal{S}_3(x)] - \frac{1}{n} \sum_{i \in [n]} y_i \cdot \mathcal{P}_3(x_i),$$

where $\widehat{T} = \frac{1}{n} \sum_{i \in [n]} y_i \cdot \mathcal{P}_3(x_i)$ is the empirical form used in NN-LIFT Algorithm 6. Note that the observations are from the arbitrary function $y = f(x)$. The perturbation tensor can be expanded as

$$E = \underbrace{\mathbb{E}[\tilde{y}_f \cdot \mathcal{S}_3(x)] - \mathbb{E}[y \cdot \mathcal{S}_3(x)]}_{:= E_{\text{apx.}}} + \underbrace{\mathbb{E}[y \cdot \mathcal{S}_3(x)] - \frac{1}{n} \sum_{i \in [n]} y_i \cdot \mathcal{P}_3(x_i)}_{:= E_{\text{est.}}},$$

where $E_{\text{apx.}}$ and $E_{\text{est.}}$ respectively denote the perturbations from approximation and estimation parts.

We also desire to use the exact second order moment $\tilde{M}_{2,f} = \mathbb{E}[\tilde{y}_f \cdot \mathcal{S}_2(x)]$ for the whitening Procedure 10 in the tensor decomposition method. But, we have an empirical version for which the perturbation matrix $E_2 := \tilde{M}_{2,f} - \widehat{M}_2$ is expanded as

$$E_2 = \underbrace{\mathbb{E}[\tilde{y}_f \cdot \mathcal{S}_2(x)] - \mathbb{E}[y \cdot \mathcal{S}_2(x)]}_{:= E_{2,\text{apx.}}} + \underbrace{\mathbb{E}[y \cdot \mathcal{S}_2(x)] - \frac{1}{n} \sum_{i \in [n]} y_i \cdot \mathcal{P}_2(x_i)}_{:= E_{2,\text{est.}}},$$

where $E_{2,\text{apx.}}$ and $E_{2,\text{est.}}$ respectively denote the perturbations from approximation and estimation parts.

In Theorem 4.3 where there is no approximation error, we only need to analyze the estimation perturbations characterized in (D.4) and (D.5) since the neural network output is directly observed (and thus, we use $\tilde{y}$ to denote the output). Now, the goal is to argue that the norm of perturbations E and E_2 are small enough (see Lemma D.2), ensuring the tensor power iteration recovers the rank-1 components of $\tilde{T}_f = \mathbb{E}[\tilde{y}_f \cdot \mathcal{S}_3(x)]$ with bounded error. Again recall from Lemma 4.1 that the rank-1 components of tensor $\tilde{T}_f = \mathbb{E}[\tilde{y}_f \cdot \mathcal{S}_3(x)]$ are the desired components to recover.

The estimation perturbations $E_{\text{est.}}$ and $E_{2,\text{est.}}$ are similarly bounded as in Lemma D.1 (see (D.6) and (D.7)), and thus, we have w.h.p.

$$\|E_{\text{est.}}\| \le \tilde{O}\left(\frac{y_{\max}}{\sqrt{n}}\sqrt{\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]}\right),$$
$$\|E_{2,\text{est.}}\| \le \tilde{O}\left(\frac{y_{\max}}{\sqrt{n}}\sqrt{\mathbb{E}\left[\left\|\mathcal{S}_2(x)\mathcal{S}_2^\top(x)\right\|\right]}\right),$$

where $M_3(x) \in \mathbb{R}^{d\times d^2}$ denotes the matricization of score function tensor $\mathcal{S}_3(x) \in \mathbb{R}^{d\times d\times d}$, and $y_{\max}$ is the bound on $|f(x)| = |y|$.

The norm of approximation perturbation $E_{\text{apx.}} := \mathbb{E}[(\tilde{y}_f - y)\cdot \mathcal{S}_3(x)]$ is bounded as

$$\begin{aligned}
\|E_{\text{apx.}}\| &= \|\mathbb{E}[(\tilde{y}_f - y)\cdot \mathcal{S}_3(x)]\| \\
&\le \mathbb{E}[\|(\tilde{y}_f - y)\cdot \mathcal{S}_3(x)\|] \\
&= \mathbb{E}[|\tilde{y}_f - y|\cdot \|\mathcal{S}_3(x)\|] \\
&\le \left(\mathbb{E}[|\tilde{y}_f - y|^2]\cdot \mathbb{E}[\|\mathcal{S}_3(x)\|^2]\right)^{1/2},
\end{aligned}$$

where the first inequality is from the Jensen's inequality applied to convex norm function, and we used Cauchy-Schwartz in the last inequality. Applying the approximation bound in Theorem 4.4, we have

$$\|E_{\text{apx.}}\| \le O(rC_f)\cdot\left(\frac{1}{\sqrt{k}} + \delta_1\right)\cdot\sqrt{\mathbb{E}[\|\mathcal{S}_3(x)\|^2]}, \tag{D.32}$$

and similarly,

$$\|E_{2,\text{apx.}}\| \le O(rC_f)\cdot\left(\frac{1}{\sqrt{k}} + \delta_1\right)\cdot\sqrt{\mathbb{E}[\|\mathcal{S}_2(x)\|^2]},$$

We now need to ensure the overall perturbations $E = E_{\text{est.}} + E_{\text{apx.}}$ and $E_2 = E_{2,\text{est.}} + E_{2,\text{apx.}}$ satisfies the required bounds in Lemma D.2. Note that similar to what we do in Lemma D.1, we first impose a bound such that the term involving $\|E\|$ is dominant in (D.17). Bounding the estimation part $\|E_{\text{est.}}\|$ provides similar sample complexity as in estimation Lemma D.1 with $\tilde{y}_{\max}$ substituted by $y_{\max}$.

For the approximation error, by imposing (third bound stated in the theorem)

$$C_f \leq O\left(\frac{1}{r}\left(\frac{1}{\sqrt{k}}+\delta_1\right)^{-1}\frac{\mathbb{E}[\|\mathcal{S}_3(x)\|^2]^{1/4}}{\mathbb{E}[\|\mathcal{S}_2(x)\|^2]^{3/4}}\cdot\tilde{\lambda}_{\min}\cdot s^{1.5}_{\min}(A_1)\right),$$

we ensure that the term involving $\|E\|$ is dominant in the final recovery error in (D.17). By doing this, we do not need to impose the bound on $\|E_{2,\text{apx.}}\|$ anymore, and applying the bound in (D.32) to the required bound on $\|E\|$ in Lemma D.2 leads to bound (second bound stated in the theorem)

$$C_f \leq \tilde{O}\left(\frac{1}{r}\left(\frac{1}{\sqrt{k}}+\delta_1\right)^{-1}\frac{1}{\sqrt{\mathbb{E}[\|\mathcal{S}_3(x)\|^2]}}\cdot\lambda_{\min}\left(\frac{\tilde{\lambda}_{\min}}{\tilde{\lambda}_{\max}}\right)^{1.5}s^3_{\min}(A_1)\cdot\frac{1}{\sqrt{k}}\right).$$

Finally, applying the result of Lemma D.2, we have the column-wise error guarantees (up to permutation)

$$\begin{aligned}
\|(A_1)_j-(\widehat{A}_1)_j\| &\leq \tilde{O}\left(\frac{s_{\max}(A_1)}{\lambda_{\min}}\frac{\tilde{\lambda}^2_{\max}}{\sqrt{\tilde{\lambda}_{\min}}}\frac{\|E_{\text{est.}}\|+\|E_{\text{apx.}}\|}{\tilde{\lambda}^{1.5}_{\min}\cdot s^3_{\min}(A_1)}\right),\\
&\leq \tilde{O}\left(\frac{\tilde{\lambda}^2_{\max}}{\tilde{\lambda}^2_{\min}}\frac{s_{\max}(A_1)}{\lambda_{\min}\cdot s^3_{\min}(A_1)}\left[\frac{y_{\max}}{\sqrt{n}}\sqrt{\mathbb{E}\left[\left\|M_3(x)M_3^\top(x)\right\|\right]}\right.\right.\\
&\qquad\qquad\left.\left.+\ rC_f\cdot\left(\frac{1}{\sqrt{k}}+\delta_1\right)\cdot\sqrt{\mathbb{E}[\|\mathcal{S}_3(x)\|^2]}\right]\right)\\
&\leq \tilde{O}\left(\tilde{\epsilon}_1\right),
\end{aligned}$$

where in the second inequality we substituted the earlier bounds on $\|E_{\text{est.}}\|$ and $\|E_{\text{apx.}}\|$, and the first bounds on n and C_f stated in the theorem are used in the last inequality.

Fourier part: Let

$$\tilde{v}_f := \frac{1}{n}\sum_{i\in[n]}\frac{(\tilde{y}_f)_i}{p(x_i)}e^{-j\langle\omega_i,x_i\rangle}.$$

Note that this a realization of $\tilde{v}$ defined in (D.22) when the output is generated by a neural network satisfying approximation error provided in Theorem 4.4 denoted by $\tilde{y}_f$; see the discussion in the beginning of the proof.

The perturbation is now

$$e := \mathbb{E}[\tilde{v}_f] - \underbrace{\frac{1}{n}\sum_{i\in[n]} \frac{y_i}{p(x_i)} e^{-j\langle \omega_i, x_i\rangle}}_{=:v}.$$

Similar to the tensor decomposition part, it can be expanded to estimation and approximation parts as

$$e := \underbrace{\mathbb{E}[\tilde{v}_f] - \mathbb{E}[v]}_{e_{\text{apx.}}} + \underbrace{\mathbb{E}[v] - v}_{e_{\text{est.}}}.$$

Similar to Lemma D.3, the estimation error is w.h.p. bounded as

$$|e_{\text{est.}}| \leq O(\tilde{\epsilon}_2),$$

if the sample complexity satisfies $n \geq \tilde{O}\left(\frac{\zeta_f}{\psi\tilde{\epsilon}_2^2}\right)$, where $\zeta_f := \int_{\mathbb{R}^d} f(x)^2 dx$. Notice the difference between ζ_f and $\tilde{\zeta}_{\tilde{f}}$. The approximation part is also bounded as

$$|e_{\text{apx.}}| \leq \frac{1}{\psi}\mathbb{E}[|\tilde{y}_f - y|] \leq \frac{1}{\psi}\sqrt{\mathbb{E}[|\tilde{y}_f - y|^2]} \leq \frac{1}{\psi} O(rC_f) \cdot \left(\frac{1}{\sqrt{k}} + \delta_1\right),$$

where the last inequality is from the approximation bound in Theorem 4.4. Imposing the condition

$$C_f \leq \frac{1}{r}\left(\frac{1}{\sqrt{k}} + \delta_1\right)^{-1} \cdot O(\psi\tilde{\epsilon}_2) \tag{D.33}$$

satisfies the desired bound $|e_{\text{apx.}}| \leq O(\tilde{\epsilon}_2)$. The rest of the analysis is the same as Lemma D.3.

Ridge regression: It introduces an additional approximation term in the linear regression formulated in (D.27). Given the above bounds on C_f, the new approximation term only contributes to lower order terms.

Combining the analyzes for tensor decomposition, Fourier and ridge regression parts finishes the proof. □

D.3.1 Discussion on Corollary 4.1

Similar to the specific Gaussian kernel function, we can also provide the results for other kernel functions and in general for positive definite functions as follows. $f(x)$ is said to be positive definite if $\sum_{j,l} x_i x_j f(x_j - x_l) \geq 0$, for all $x_j, x_l \in \mathbb{R}^d$. Barron [35] shows that positive definite functions have C_f bounded as

$$C_f \leq \sqrt{-f(0) \cdot \nabla^2 f(0)},$$

where $\nabla^2 f(x) := \sum_{i \in [d]} \partial^2 f(x)/\partial x_i^2$. Note that the operator ∇^2 is different from the derivative operator $\nabla^{(2)}$ that we defined in (4.1). Applying this to the proposed bound in (4.15), we conclude that our algorithm can train a neural network which approximates a class of positive definite and kernel functions with similar bounds as in Theorem 4.5. Corollary 4.1 is for the special case of Gaussian kernel function.

Proof of Corollary 4.1: For the location and scale mixture $f(x) := \int K(\alpha(x+\beta))G(d\alpha, d\beta)$, we have [35] $C_f \leq C_K \cdot \int |\alpha| \cdot |G|(d\alpha, d\beta)$, where C_K denotes the corresponding parameter for $K(x)$. For the standard Gaussian kernel function $K(x)$ considered here, we have [35] $C_K \leq \sqrt{d}$, which concludes

$$C_f \leq \sqrt{d} \cdot \int |\alpha| \cdot |G|(d\alpha, d\beta).$$

We now apply the required bound in (4.15) to finish the proof. But, in this specific setting, we also have the following simplifications for the bound in (4.15).

For the Gaussian input $x \sim \mathcal{N}(0, \sigma_x^2 I_d)$, the score function is

$$\mathcal{S}_3(x) = \frac{1}{\sigma_x^6} x^{\otimes 3} - \frac{1}{\sigma_x^4} \sum_{j \in [d]} \left(x \otimes e_j \otimes e_j + e_j \otimes x \otimes e_j + e_j \otimes e_j \otimes x \right),$$

which has expected square norm as $\mathbb{E}[\|\mathcal{S}_3(x)\|^2] = \tilde{O}(d^3/\sigma_x^6)$.

Given the input is Gaussian and the activating function is the step function, we can also write the coefficients λ_j and $\tilde{\lambda}_j$ as

$$\lambda_j = a_2(j) \cdot \frac{1}{\sqrt{2\pi}\sigma_x^3} \cdot \exp\left(-\frac{b_1(j)^2}{2\sigma_x^2}\right) \cdot \left(\frac{b_1(j)^2}{\sigma_x^2} - 1\right),$$
$$\tilde{\lambda}_j = a_2(j) \cdot \frac{b_1(j)}{\sqrt{2\pi}\sigma_x^3} \cdot \exp\left(-\frac{b_1(j)^2}{2\sigma_x^2}\right).$$

Given the bounds on coefficients as $|b_1(j)| \leq 1$, $|a_2(j)| \leq 2C_f$, $j \in [k]$, we have

$$\frac{\tilde{\lambda}_{\min}}{\tilde{\lambda}_{\max}} \geq \frac{(a_2)_{\min} \cdot (b_1)_{\min}}{2C_f} \exp(-1/(2\sigma_x^2)),$$
$$\lambda_{\min} \geq \frac{(a_2)_{\min}}{\sqrt{2\pi}\sigma_x^3} \exp(-1/(2\sigma_x^2)) \cdot \min_{j\in[k]} |b_1(j)^2/\sigma_x^2 - 1|.$$

Recall that the columns of A_1 are randomly drawn from the Fourier spectrum of $f(x)$ as described in (4.12). Given $f(x)$ is the Gaussian kernel, the Fourier spectrum $\|\omega\| \cdot |F(\omega)|$ corresponds to a sub-gaussain distribution. Thus, the singular values of A_1 are bounded as [139]

$$\frac{s_{\min}(A_1)}{s_{\max}(A_1)} \geq \frac{1 - \sqrt{k/d}}{1 + \sqrt{k/d}} \geq O(1),$$

where the last inequality is from $k = Cd$ for some small enough $C < 1$.

Substituting these bounds in the required bound in (4.15) finishes the proof. □

Appendix E

Proofs for Identifiability of Overcomplete Topic Models

E.1 Proof of Deterministic Identifiability Result (Theorem 5.1)

First, we show the identifiability result under an alternative set of conditions on the n-gram matrix, $A^{\odot n}$, and then, we show that the conditions of Theorem 5.1 are sufficient for these conditions to hold.

E.1.1 Deterministic Analysis Based on $A^{\odot n}$

In this section, the deterministic identifiability result based on conditions on the n-gram matrix, $A^{\odot n}$, is provided.

In the n-gram matrix, $A^{\odot n} \in \mathbb{R}^{p^n \times q}$, redundant rows exist. If some row of $A^{\odot n}$ is indexed by n-tuple $(i_1, \ldots, i_n) \in [p]^n$, then another row indexed by any permutation of the tuple $(i_1, \ldots, i_n)$ has the same entries. Therefore, the number of distinct rows of $A^{\odot n}$ is at most $\binom{p+n-1}{n}$. In the following definition, we define a non-redundant version of n-gram matrix which is restricted to the (potentially) distinct rows.

Definition E.1 (Restricted n-gram matrix)**.** *For any matrix $A \in \mathbb{R}^{p\times q}$, restricted n-gram matrix $A^{\odot n}_{\text{Rest.}} \in \mathbb{R}^{s\times q}$, $s = \binom{p+n-1}{n}$, is defined as the restricted version of n-gram matrix $A^{\odot n} \in \mathbb{R}^{p^n\times q}$, where the redundant rows of $A^{\odot n}$ are removed, as explained above.*

Condition 6 (Rank condition)**.** *The n-gram matrix $A^{\odot n}$ is full column rank.*

Condition 7 (Graph expansion)**.** *Let $G(V_h, V_o^{(n)}; A^{\odot n})$ denote the bipartite graph with vertex sets V_h corresponding to the hidden variables (indexing the columns of $A^{\odot n}$) and $V_o^{(n)}$ corresponding to the n-th order observed variables (indexing the rows of $A^{\odot n}$) and edge matrix $A^{\odot n} \in \mathbb{R}^{|V_o^{(n)}|\times|V_h|}$. The bipartite graph $G(V_h, V_o^{(n)}; A^{\odot n})$ satisfies the following* expansion property [1] *on the restricted version specified by $A^{\odot n}_{\text{Rest.}}$,*

$$\left|N_{A^{\odot n}_{\text{Rest.}}}(S)\right| \geq |S| + d_{\max}\left(A^{\odot n}\right), \quad \forall S \subseteq V_h,\ |S| > \text{krank}(A), \tag{E.1}$$

where $d_{\max}\left(A^{\odot n}\right)$ is the maximum node degree in set V_h.

Remark 31. The expansion condition for the bag-of-words admixture model is provided in (5.4), introduced in [9]. The proposed expansion condition in (E.1) is inherited from (5.4), with two major modifications. First, the condition is appropriately generalized for our model which involves a graph with edges specified by the n-gram matrix, $A^{\odot n}$, as stated in (5.23). Second, the expansion property (5.4), proposed in [9], needs to be satisfied for all subsets S with size $|S| \geq 2$, which is a stricter condition than the one proposed here in (E.1), since we can have $\text{krank}(A) \gg 2$.

The deterministic identifiability result based on the conditions on $A^{\odot n}$, is stated in the following theorem for $n \geq 2$, while $n = 1$ case is addressed in Remarks 19 and 31. The identifiability result relies on access to the $(2n)$-th order moment of observed variables $x_l, l \in [2n]$, defined in equation (5.2) as

$$M_{2n}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_n)(x_{n+1} \otimes x_{n+2} \otimes \cdots \otimes x_{2n})^\top\right] \in \mathbb{R}^{p^n\times p^n}.$$

[1] Note that this notion of generalized expansion is different from unbalanced expander graphs proposed in the compressed sensing literature [106, 100]. For a left regular bipartite graph $G(Y, X; A)$ with regular degree d for the vertices on Y side, we say that it is a (k, ϵ)-expander if for any set $S \subseteq Y$ with $|S| \leq k$, we have $N_A(S) \geq |S|d(1-\epsilon)$. This is completely different with the expansion condition we define here in some aspects: first our expansion condition is additive while this one is multiplicative, and second our expansion condition is imposed on large sets while this one is imposed on small sets.

Theorem E.1 (Generic identifiability under deterministic conditions on $A^{\odot n}$)**.** *Let $M_{2n}^{(n)}(x)$ (defined in equation (5.2)) be the $(2n)$-th order moment of the n-persistent topic model described in Section 5.4. If the model satisfies conditions 1, 6 and 7, then, for any $n \geq 2$, all the columns of population structure A are* generically identifiable *from $M_{2n}^{(n)}(x)$.*

Proof: Define $B := A^{\odot n} \in \mathbb{R}^{p^n \times q}$. Then, the moment characterized in equation (5.23) can be written as $M_{2n}^{(n)}(x) = B\mathbb{E}\left[hh^\top\right]B^\top$. Since both matrices $\mathbb{E}\left[hh^\top\right]$ and B have full column rank (from conditions 1 and 6), the rank of $B\mathbb{E}\left[hh^\top\right]B^\top$ is q where $q = O(p^n)$, and furthermore $\mathrm{Col}(B\mathbb{E}\left[hh^\top\right]B^\top) = \mathrm{Col}(B)$. Let $\mathcal{U} := \{u_1, \ldots, u_q\} \in \mathbb{R}^{p^n}$ be any basis of $\mathrm{Col}(B\mathbb{E}\left[hh^\top\right]B^\top)$ satisfying the following two properties:

1) The maximum of ℓ_0 norm of u_i's is minimized (among all basis sets).

2) The tensor rank of u_i's (in the n-th order tensor form) is equal to 1, i.e., $\mathrm{Rank}(\mathrm{ten}(u_i)) = 1, i \in [q]$.

Let the columns of matrix B be b_i for $i \in [q]$. Since all the b_i's (which belong to $\mathrm{Col}(B\mathbb{E}\left[hh^\top\right]B^\top)$) are rank-1 in the n-th order tensor form (since $\mathrm{ten}(b_i) = a_i^{\otimes n}$) and the number of non-zero entries in each of b_i's is at most $d_{\max}(B) = d_{\max}(A)^n$, we conclude that

$$\max_i \mathrm{Rank}(\mathrm{ten}(u_i)) = 1 \quad \text{and} \quad \max_i \|u_i\|_0 \leq d_{\max}(B). \tag{E.2}$$

The above bounds are concluded from the fact that $b_i \in \mathrm{Col}(B\mathbb{E}\left[hh^\top\right]B^\top),\ i \in [q]$, and therefore the ℓ_0 norm and the rank properties of b_i's are upper bounds for the corresponding properties of basis vectors u_i's (according to the proposed conditions for u_i's).

Now, exploiting these observations and also the genericity of A and the expansion condition 7, we show that the basis vectors u_i's are scaled columns of B. Since u_i for $i \in [q]$, is a vector in the column space of B, it can be represented as $u_i = Bv_i$ for some vector $v_i \in \mathbb{R}^q$. Equivalently, for any $i \in [q]$, $u_i = \sum_{j=1}^q v_i(j)b_j$ where $b_j = a_j^{\otimes n}$ is the j-th column of matrix B and $v_i(j)$ is a scalar

which is the j-th entry of vector v_i. Then, the tensor form of u_i can be written as

$$\text{ten}(u_i) = \sum_{j=1}^{q} v_i(j)\,\text{ten}(b_j) = \sum_{j=1}^{q} v_i(j)\,\text{ten}(a_j^{\otimes n}) = \sum_{j=1}^{q} v_i(j) a_j^{\circ n} = [[\text{Diag}_n(v_i); \overbrace{A, \ldots, A}^{n \text{ times}}]], \quad \text{(E.3)}$$

where the last equality is based on the notation defined in Definition 5.8, and $\text{Diag}_n(v_i)$ is defined as the n-th order diagonal tensor with vector v_i on its diagonal. We define $\widetilde{v}_i := [v_i(j)]_{j:v_i(j)\neq 0}$ as the vector which contains only the non-zero entries of v_i, i.e., $\widetilde{v}_i$ is the restriction of vector v_i to its support. Therefore, $\widetilde{v}_i \in \mathbb{R}^r$, where $r := \|v_i\|_0$. Furthermore, the matrix $\widetilde{A}_i := \{a_j : v_i(j) \neq 0\} \in \mathbb{R}^{p\times r}$ is defined as the restriction of A to its columns corresponding to the support of v_i. Let $(\widetilde{a}_i)_j$ denote the j-th column of $\widetilde{A}_i$. According to these definitions, equation (E.3) reduces to

$$\text{ten}(u_i) = [[\text{Diag}_n(\widetilde{v}_i); \overbrace{\widetilde{A}_i, \ldots, \widetilde{A}_i}^{n \text{ times}}]] = \sum_{j=1}^{r} \widetilde{v}_i(j)[(\widetilde{a}_i)_j]^{\circ n}, \quad \text{(E.4)}$$

which is derived by removing columns of A corresponding to the zero entries in v_i.
Next, we rule out that $\|v_i\|_0 \geq 2$ under two cases ($2 \leq \|v_i\|_0 \leq \text{krank}(A)$ and $\text{krank}(A) < \|v_i\|_0 \leq q$), to conclude that u_i's vectors are scaled columns of B.

Case 1: $2 \leq \|v_i\|_0 \leq \text{krank}(A)$**.** Here, the number of columns of $\widetilde{A}_i \in \mathbb{R}^{p\times\|v_i\|_0}$ is less than or equal to $\text{krank}(A)$ and therefore it is full column rank. Since, all the components of CP representation in equation (E.4) are full column rank [2], for any [3] $n \geq 2$, we have $\text{Rank}(\text{ten}(u_i)) = r = \|v_i\|_0 > 1$, which contradicts the fact that $\max_i \text{Rank}(\text{ten}(u_i)) = 1$ in (E.2).

Note that for the *full column rank* topic-word matrix $A \in \mathbb{R}^{p\times q}$ (where $\text{Rank}(A) = \text{krank}(A) = q$) as in Corollary 5.1, it is sufficient to argue this case and there is no need to argue next case. This is why the expansion condition is not required in Corollary 5.1.

Case 2: $\text{krank}(A) < \|v_i\|_0 \leq q$**.** Here, we first restrict the n-gram matrix B to distinct rows, denoted by $B_{\text{Rest.}}$, as defined in Definition E.1. Let $u_i' = B_{\text{Rest.}} v_i$. Since u_i' is the restricted version

[2] Note that for $n \geq 3$, this full rank condition can be relaxed by Kruskal's condition for uniqueness of CP decomposition [112] and its generalization to higher order tensors [143]. Precisely, instead of saying $\text{Rank}(\widetilde{A}_i) = \text{krank}(\widetilde{A}_i) = r$, it is only required to have $\text{krank}(\widetilde{A}_i) \geq (2r+n-1)/n$ to argue the result of case 1. This only improves the constants involved in the final result.

[3] Note that for $n = 1$, since the (tensor) rank of any vector is 1, this analysis does not work.

of u_i, we have

$$\begin{aligned}\|u_i\|_0 \geq \|u_i'\|_0 &= \|B_{\text{Rest.}} v_i\|_0 \\ &> \left|N_{B_{\text{Rest.}}}(\text{Supp}(v_i))\right| - |\,\text{Supp}(v_i)| \\ &\geq d_{\max}(B),\end{aligned}$$

where the second inequality is from Lemma 4 (which is stated and proved right after this theorem), and the third inequality follows from the graph expansion property (condition 7). This result contradicts the fact that $\max_i \|u_i\|_0 \leq d_{\max}(B)$ in (E.2).

From above contradictions, $\|v_i\|_0 = 1$ and hence, columns of $B := A^{\odot n}$ are the scaled versions of u_i's. □

The following lemma is useful in the proof of Theorem E.1. The result proposed in this lemma is similar to the parameter genericity condition in [9], but generalized for the n-gram matrix, $A^{\odot n}$. The lemma is proved along the lines of the proof of Remark 2.2 in [9].

Lemma 4. *If $A \in \mathbb{R}^{p\times q}$ is generic (see Definition 5.1), then the n-gram matrix $A^{\odot n} \in \mathbb{R}^{p^n \times q}$ satisfies the following property with Lebesgue measure one. For any vector $v \in \mathbb{R}^q$ with $\|v\|_0 \geq 2$, we have*

$$\left\|A^{\odot n}_{\text{Rest.}} v\right\|_0 > \left|N_{A^{\odot n}_{\text{Rest.}}}(\text{Supp}(v))\right| - |\,\text{Supp}(v)|,$$

where for a set $S \subseteq [q]$, $N_{A^{\odot n}}(S) := \{i \in [p]^n : A^{\odot n}(i,j) \neq 0 \text{ for some } j \in S\}$.

Here, we prove the result for the case of $n = 2$. The proof can be easily generalized to larger n.

Let $A := P + Z$ be generic, where P is an arbitrary matrix perturbed by random continuous independent[4] perturbations Z. Consider the 2-gram matrix $B := A \odot A \in \mathbb{R}^{p^2\times q}$. We show that the restricted version of B, denoted by $\widetilde{B} := B_{\text{Rest.}} \in \mathbb{R}^{\frac{p(p+1)}{2}\times q}$, satisfies the above genericity condition. Before that, we first establish some definitions and one claim.

[4]Note that the distribution of Z does not matter as long as the independence and continuous conditions hold.

Definition E.2. *We call a vector* fully dense *if all of its entries are non-zero.*

Definition E.3. *We say a matrix has the* Null Space Property (NSP) *if its null space does not contain any fully dense vector.*

Claim 20. *Fix any $S \subseteq [q]$ with $|S| \geq 2$, and set $R := N_{(P^{\odot 2})_{\text{Rest.}}}(S)$. Let $\widetilde{C}$ be a $|S| \times |S|$ submatrix of $\widetilde{B}_{R,S}$. Then* $\Pr(\widetilde{C} \text{ has the NSP}) = 1$.

Proof of Claim 20: First, note that $\widetilde{B}$ can be expanded as

$$\widetilde{B} := (A \odot A)_{\text{Rest.}} = (P \odot P)_{\text{Rest.}} + \underbrace{(P \odot Z + Z \odot P)_{\text{Rest.}} + (Z \odot Z)_{\text{Rest.}}}_{:=U}.$$

Let $s = |S|$ and let $\widetilde{C} = [\tilde{c}_1 | \tilde{c}_2 | \cdots | \tilde{c}_s]^\top$, where $\tilde{c}_i^\top$ is the i-th row of $\widetilde{C}$. Also, let $C := [c_1 | c_2 | \cdots | c_s]^\top$ and $W := [w_1 | w_2 | \cdots | w_s]^\top$ be the corresponding $|S| \times |S|$ submatrices of $\left(P^{\odot 2}\right)_{\text{Rest.}}$ and U, respectively. For each $i \in [s]$, denote by $\mathcal{N}_i$ the null space of the matrix $\widetilde{C}_i = [\tilde{c}_1 | \tilde{c}_2 | \cdots | \tilde{c}_i]^\top$. Finally let $\mathcal{N}_0 = \mathbb{R}^s$. Then, $\mathcal{N}_0 \supseteq \mathcal{N}_1 \supseteq \cdots \supseteq \mathcal{N}_s$. We need to show that, with probability one, $\mathcal{N}_s$ does not contain any fully dense vector.

If one of $\mathcal{N}_i, i \in [s]$, does not contain any full dense vector, the result is proved. Suppose that $\mathcal{N}_i$ contains some fully dense vector v. Since C is a submatrix of $\left(P^{\odot 2}\right)_{R,S}$, every row $c_{i+1}^\top$ of C contains at least one non-zero entry. Therefore,

$$\begin{aligned} v^\top \tilde{c}_{i+1} &= \sum_{j \in [s]} v(j) \tilde{c}_{i+1}(j) \\ &= \sum_{j \in [s] : c_{i+1}(j) \neq 0} v(j)(c_{i+1}(j) + w_{i+1}(j)), \end{aligned}$$

where $\{w_{i+1}(j) : j \in [s] \text{ s.t. } c_{i+1}(j) \neq 0\}$ are independent random variables, and moreover, they are independent of $\tilde{c}_1, \ldots, \tilde{c}_i$ and thus of v. By assumption on the distribution of the $w_{i+1}(j)$,

$$\Pr\left[v \in \mathcal{N}_{i+1} \middle| \tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_i \right] = \Pr\left[\sum_{j \in [s] : c_{i+1}(j) \neq 0} v(j)(c_{i+1}(j) + w_{i+1}(j)) = 0 \middle| \tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_i \right] = 0 \quad \text{(E.5)}$$

Consequently,

$$\Pr\left[\dim(\mathcal{N}_{i+1}) < \dim(\mathcal{N}_i)\middle|\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_i\right] = 1 \tag{E.6}$$

for all $i = 0, \ldots, s-1$. As a result, with probability one, $\dim(\mathcal{N}_s) = 0$. □

Now, we are ready to prove Lemma 4.

Proof of Lemma 4: It follows from Claim 20 that, with probability one, the following event holds: for every $S \subseteq [q], |S| \geq 2$, and every $|S| \times |S|$ submatrix $\widetilde{C}$ of $\widetilde{B}_{R,S}$ where $R := N_{(P^{\odot 2})_{\text{Rest.}}}(S)$, then $\tilde{C}$ has the NSP.

Now fix $v \in \mathbb{R}^q$ with $\|v\|_0 \geq 2$. Let $S := \text{Supp}(v)$ and $H := \widetilde{B}_{R,S}$. Furthermore, let $u \in (\mathbb{R} \setminus \{0\})^{|S|}$ be the restriction of vector v to S; observe that u is fully dense. It is clear that $\|\widetilde{B}v\|_0 = \|Hu\|_0$, so we need to show that

$$\|Hu\|_0 > |R| - |S|. \tag{E.7}$$

For the sake of contradiction, suppose that Hu has at most $|R| - |S|$ non-zero entries. Since $Hu \in \mathbb{R}^{|R|}$, there is a subset of $|S|$ entries on which Hu is zero. This corresponds to a $|S| \times |S|$ submatrix of $H := \widetilde{B}_{R,S}$ which contains u in its null space. It means that this submatrix does not have the NSP, which is a contradiction. Therefore we conclude that Hu must have more than $|R| - |S|$ non-zero entries, which finishes the proof. □

E.1.2 Proof of Moment Characterization Lemmata

Remark 32. In Lemmata 2 and 3, a specific case of order and persistence ($m = rn$) was considered. Here, we provide the moment form for a more general case. Assume that $m = rn + s$ for some

integers $r \geq 1, 1 \leq s \leq \frac{n}{2}$, then

$$M_{2m}^{(n)}(x) = \Big(\overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r \text{ times}} \otimes A^{\odot s} \Big) \widetilde{M}_{2r}(h) \Big(A^{\odot (n-s)} \otimes \overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r-1 \text{ times}} \otimes A^{\odot (2s)} \Big)^{\top},$$

where $\widetilde{M}_{2r}(h) \in \mathbb{R}^{q^{r+1} \times q^{r+1}}$ is the hidden moment as

$$\widetilde{M}_{2r}(h)_{\big((i_1,\ldots,i_{r+1}),(j_1,\ldots,j_{r+1})\big)} := \begin{cases} \mathbb{E}[h_{i_1} \cdots h_{i_r} h_{i_{r+1}}^2 h_{j_2} \cdots h_{j_{r+1}}] & \text{if } i_{r+1} = j_1, \\ 0 & \text{o. w .} \end{cases}$$

The tensor form is also characterized as

$$T_{2m}^{(n)}(x) = \Big[\Big[\widetilde{S}_r; \overbrace{A, A, \ldots, A}^{2m \text{ times}} \Big]\Big],$$

where $\widetilde{S}_r \in \bigotimes^{2m} \mathbb{R}^q$ is the core tensor in the above Tucker representation with the sparsity pattern as follows. Let $\mathbf{i} := (i_1, i_2, \ldots, i_{2m})$. If

$$i_1 = i_2 = \cdots = i_n, i_{n+1} = i_{n+2} = \cdots = i_{2n}, \cdots, i_{(2r-1)n+1} = i_{(2r-1)n+2} = \cdots = i_{2rn},$$
$$i_{2(m-s)+1} = i_{2(m-s)+2} = \cdots = i_{2m},$$

we have

$$\widetilde{S}_r(\mathbf{i}) = \widetilde{M}_{2r}(h)_{\big((i_n, i_{2n}, \ldots, i_{rn}, i_m), (i_{(r+1)n}, i_{(r+2)n}, \ldots, i_{2rn}, i_{2m})\big)}.$$

Otherwise, $\widetilde{S}_r(\mathbf{i}) = 0$.

Proof of Lemma 2: The proof is basically incorporating the conditional independence relationships between random variables x_l and y_j under the n-persistent topic model.

In order to simplify the notation, similar to tensor powers for vectors, the tensor power for a matrix $U \in \mathbb{R}^{p\times q}$ is defined as

$$U^{\otimes r} := \overbrace{U \otimes U \otimes \cdots \otimes U}^{r \text{ times}} \in \mathbb{R}^{p^r \times q^r}. \tag{E.8}$$

First, consider the case $m = rn$ for some integer $r \geq 1$. One advantage of encoding $y_j, j \in [2r]$, by basis vectors appears in characterizing the conditional moments. The first order conditional moment of words $x_l, l \in [2m]$, in the n-persistent topic model can be written as

$$\mathbb{E}\big[x_{(j-1)n+k}|y_j\big] = Ay_j,\ j \in [2r],\ k \in [n],$$

where $A = [a_1|a_2|\cdots|a_q] \in \mathbb{R}^{p\times q}$. Next, the m-th order conditional moment of different views $x_l, l \in [m]$, in the n-persistent topic model can be written as

$$\mathbb{E}[x_1 \otimes x_2 \otimes \cdots \otimes x_m|y_1 = e_{i_1}, y_2 = e_{i_2}, \ldots, y_r = e_{i_r}] = a_{i_1}^{\otimes n} \otimes a_{i_2}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n},$$

which is derived from the conditional independence relationships among the observations $x_l, l \in [m]$, given topics $y_j, j \in [r]$. Similar to the first order moments, since vectors $y_j, j \in [r]$, are encoded by the basis vectors $e_i \in \mathbb{R}^q$, the above moment can be written as the following matrix multiplication

$$\mathbb{E}[x_1 \otimes x_2 \otimes \cdots \otimes x_m|y_1, y_2, \ldots, y_r] = \left(A^{\odot n}\right)^{\otimes r} (y_1 \otimes y_2 \otimes \cdots \otimes y_r), \tag{E.9}$$

where the $(\cdot)^{\otimes r}$ notation is defined in equation (E.8). Now for the $(2m)$-th order moment, we have

$$
\begin{aligned}
M_{2m}^{(n)}(x) &:= \mathbb{E}\Big[(x_1 \otimes x_2 \otimes \cdots \otimes x_m)(x_{m+1} \otimes x_{m+2} \otimes \cdots \otimes x_{2m})^\top\Big] \\
&= \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\Big[\mathbb{E}\left[(x_1 \otimes \cdots \otimes x_m)(x_{m+1} \otimes \cdots \otimes x_{2m})^\top | y_1, y_2, \ldots, y_{2r}\right]\Big] \\
&\overset{(a)}{=} \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\Big[\mathbb{E}\big[(x_1 \otimes \cdots \otimes x_m)|y_1, \ldots, y_{2r}\big]\mathbb{E}\big[(x_{m+1} \otimes \cdots \otimes x_{2m})^\top | y_1, \ldots, y_{2r}\big]\Big] \\
&\overset{(b)}{=} \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\Big[\mathbb{E}\big[(x_1 \otimes \cdots \otimes x_m)|y_1, \ldots, y_r\big]\mathbb{E}\big[(x_{m+1} \otimes \cdots \otimes x_{2m})^\top | y_{r+1}, \ldots, y_{2r}\big]\Big] \\
&\overset{(c)}{=} \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\left[\left(\Big[A^{\odot n}\Big]^{\otimes r}\right)(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top \left(\Big[A^{\odot n}\Big]^{\otimes r}\right)^\top\right] \\
&= \left(\Big[A^{\odot n}\Big]^{\otimes r}\right)\mathbb{E}\left[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top\right]\left(\Big[A^{\odot n}\Big]^{\otimes r}\right)^\top \\
&\overset{(d)}{=} \left(\Big[A^{\odot n}\Big]^{\otimes r}\right)M_{2r}(y)\left(\Big[A^{\odot n}\Big]^{\otimes r}\right)^\top, \qquad \text{(E.10)}
\end{aligned}
$$

where (a) results from the independence of $(x_1, \ldots, x_m)$ and $(x_{m+1}, \ldots, x_{2m})$ given $(y_1, y_2, \ldots, y_{2r})$ and (b) is concluded from the independence of $(x_1, \ldots, x_m)$ and $(y_{r+1}, \ldots, y_{2r})$ given $(y_1, \ldots, y_r)$ and the independence of $(x_{m+1}, \ldots, x_{2m})$ and $(y_1, \ldots, y_r)$ given $(y_{r+1}, \ldots, y_{2r})$. Equation (E.9) is used in (c) and finally, the $(2r)$-th order moment of $(y_1, \ldots, y_{2r})$ is defined as $M_{2r}(y) := \mathbb{E}\left[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top\right]$ in (d).

For $M_{2r}(y)$, we have by the law of total expectation

$$
\begin{aligned}
M_{2r}(y) &:= \mathbb{E}\big[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top\big] \\
&= \mathbb{E}_h\Big[\mathbb{E}\big[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top | h\big]\Big] \\
&= \mathbb{E}_h\left[\Big(\overbrace{h \otimes \cdots \otimes h}^{r \text{ times}}\Big)\Big(\overbrace{h \otimes \cdots \otimes h}^{r \text{ times}}\Big)^\top\right] \\
&= M_{2r}(h),
\end{aligned}
$$

where the third equality is concluded from the conditional independence of variables $y_j, j \in [2r]$, given h and the model assumption that $\mathbb{E}\big[y_j|h\big] = h, j \in [2r]$. Substituting this in equation (E.10), finishes the proof for the n-persistent topic model. Similarly, the moment of single topic model (infinite persistence) can be also derived. □

Proof of Lemma 3: Defining $\Lambda := M_{2r}(h) \in \mathbb{R}^{q^r \times q^r}$ and $B := \big[A^{\odot n}\big]^{\otimes r} \in \mathbb{R}^{p^{rn} \times q^r}$, the $(2rn)$-th order moment $M_{2rn}^{(n)}(x) \in \mathbb{R}^{p^{rn} \times p^{rn}}$ of the n-persistent topic model proposed in equation (5.8) can be written as

$$M_{2rn}^{(n)}(x) = B\Lambda B^\top.$$

Let $b_{(i_1,\ldots,i_r)} \in \mathbb{R}^{p^{rn}}$ denote the corresponding column of B indexed by r-tuple $(i_1,\ldots,i_r), i_k \in [q], k \in [r]$. Then, the above matrix equation can be expanded as

$$\begin{aligned} M_{2rn}^{(n)}(x) &= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) b_{(i_1,\ldots,i_r)} b_{(j_1,\ldots,j_r)}^\top \\ &= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) [a_{i_1}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n}][a_{j_1}^{\otimes n} \otimes \cdots \otimes a_{j_r}^{\otimes n}]^\top, \end{aligned}$$

where relation $b_{(i_1,\ldots,i_r)} = a_{i_1}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n}, i_1,\ldots,i_r \in [q]$, is used in the last equality. Let $m_{2rn}^{(n)}(x) \in \mathbb{R}^{p^{2rn}}$ denote the vectorized form of $(2rn)$-th order moment $M_{2rn}^{(n)}(x) \in \mathbb{R}^{p^{rn} \times p^{rn}}$. Therefore, we have

$$\begin{aligned} m_{2rn}^{(n)}(x) &:= \operatorname{vec}\Big(M_{2rn}^{(n)}(x)\Big) \\ &= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) a_{i_1}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n} \otimes a_{j_1}^{\otimes n} \otimes \cdots \otimes a_{j_r}^{\otimes n}. \end{aligned}$$

Then, we have the following equivalent tensor form for the original model proposed in equation (5.8)

$$\begin{aligned} T_{2rn}^{(n)}(x) &:= \operatorname{ten}\Big(m_{2rn}^{(n)}(x)\Big) \\ &= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) a_{i_1}^{\circ n} \circ \cdots \circ a_{i_r}^{\circ n} \circ a_{j_1}^{\circ n} \circ \cdots \circ a_{j_r}^{\circ n}. \end{aligned}$$

□

E.1.3 Sufficient Matching Properties for Rank and Graph Expansion Conditions

In the following lemma, it is shown that under a perfect n-gram matching and additional genericity and krank conditions, the rank and graph expansion conditions 6 and 7 on $A^{\odot n}$, are satisfied.

Lemma 5. *Assume that the bipartite graph $G(V_h, V_o; A)$ has a perfect n-gram matching (condition 2 is satisfied). Then, the following results hold for the n-gram matrix $A^{\odot n}$:*

1) If A is generic, $A^{\odot n}$ is full column rank (condition 6) with Lebesgue measure one (almost surely).

2) If krank condition 3 holds, $A^{\odot n}$ satisfies the proposed expansion property in condition 7.

Proof: Let M denote the perfect n-gram matching of the bipartite graph $G(V_h, V_o; A)$. From Lemma 1, there exists a perfect matching $M^{\odot n}$ for the bipartite graph $G(V_h, V_o^{(n)}; A^{\odot n})$. Denote the corresponding bi-adjacency matrix to the edge set M as A_M. Similarly, B_M denotes the corresponding bi-adjacency matrix to the edge set $M^{\odot n}$. Note that $\text{Supp}(A_M) \subseteq \text{Supp}(A)$ and $\text{Supp}(B_M) \subseteq \text{Supp}(A^{\odot n})$.

Since B_M is a perfect matching, it consists of $q := |V_h|$ rows, each of which has only one non-zero entry, and furthermore, the non-zero entries are in q different columns. Therefore, these rows form q linearly independent vectors. Since the row rank and column rank of a matrix are equal, and the number of columns of B_M is q, the column rank of B_M is q or in other words, B_M is full column rank. Since A is generic, from Lemma 6 (with a slight modification in the analysis [5]), $A^{\odot n}$ is also full column rank with Lebesgue measure one (almost surely). This completes the proof of part 1.

Next, we prove the second part. From krank definition, we have

$$|N_A(S')| \geq |S'| \quad \text{for } S' \subseteq V_h, |S'| \leq \text{krank}(A),$$

[5] The Lemma 6 result is about the column rank of A itself, but here it is about the column rank of $A^{\odot n}$ for which the same analysis works. Note that the support of B_M (which is full column rank here) is within the support of $A^{\odot n}$ and therefore Lemma 6 can still be applied.

which is concluded from the fact that the corresponding submatrix of A specified by S' should be full column rank. From this inequality, we have

$$|N_A(S')| \geq \text{krank}(A) \quad \text{for } S' \subseteq V_h, |S'| = \text{krank}(A). \tag{E.11}$$

Then, we have

$$\begin{aligned} |N_A(S)| &\geq |N_A(S')| \quad \text{for } S' \subset S \subseteq V_h, |S| > \text{krank}(A), |S'| = \text{krank}(A), \\ &\geq \text{krank}(A) \\ &\geq d_{\max}(A)^n, \end{aligned} \tag{E.12}$$

where (E.11) is used in the second inequality and the last inequality is from krank condition 3.

In the restricted n-gram matrix $A^{\odot n}_{\text{Rest.}}$, the number of neighbors for a set $S \subseteq V_h, |S| > \text{krank}(A)$, can be bounded as

$$\begin{aligned} \left|N_{A^{\odot n}_{\text{Rest.}}}(S)\right| &\geq |N_A(S)| + |S| \\ &\geq d_{\max}(A)^n + |S| \quad \text{for } |S| > \text{krank}(A), \end{aligned} \tag{E.13}$$

where the first inequality is due to the fact that the set $N_{A^{\odot n}_{\text{Rest.}}}$ consists of rows indexed by the following two [6] subsets: n-tuples $(i, i, \ldots, i)$ where all the indices are equal and n-tuples $(i_1, \ldots, i_n)$ with distinct indices, i.e., $i_1 \neq i_2 \ldots \neq i_n$. The former subset is exactly $N_A(S)$ while the size of the latter subset is at least $|S|$ due to the existence of a perfect n-gram matching in A. The bound (E.12) is used in the second inequality. Since $d_{\max}\left(A^{\odot n}\right) = d_{\max}(A)^n$, the proof of part 2 is also completed.

□

Remark 33. The second result of above lemma is similar to the necessity argument of (Hall's) Theorem E.2 for the existence of perfect matching in a bipartite graph, but generalized to the case of perfect n-gram matching and with additional krank condition.

[6] Note that many terms in this bound are ignored which leads to a loose bound that might be improved.

E.1.4 Auxiliary Lemma

Proof of Lemma 1: We show that if $G(Y, X; A)$ has a perfect n-gram matching, then $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching. The reverse can be also immediately shown by reversing the discussion and exploiting the additional condition stated in the lemma.

Let $E^{\odot n}$ denote the edge set of the bipartite graph $G(Y, X^{(n)}; A^{\odot n})$. Assume $G(Y, X; A)$ has a perfect n-gram matching $M \subseteq E$. For any $j \in Y$, let $N_M(j)$ denote the set of neighbors of vertex j according to edge set M. Since M is a perfect n-gram matching, $|N_M(j)| = n$ for all $j \in Y$. It can be immediately concluded from Definition 5.3 that sets $N_M(j)$ are all distinct, i.e., $N_M(j_1) \neq N_M(j_2)$ for any $j_1, j_2 \in Y, j_1 \neq j_2$. For any $j \in Y$, let $N'_M(j)$ denote an arbitrary ordered n-tuple generated from the elements of set $N_M(j)$. From the definition of n-gram matrix, we have $A^{\odot n}(N'_M(j), j) \neq 0$ for all $j \in Y$. Hence, $(j, N'_M(j)) \in E^{\odot n}$ for all $j \in Y$ which together with the fact that all $N'_M(j)$'s tuples are distinct, it results that $M^{\odot n} := \{(j, N'_M(j)) | j \in Y\} \subseteq E^{\odot n}$ is a perfect matching for $G(Y, X^{(n)}; A^{\odot n})$. □

Lemma 6. *Consider matrix $C \in \mathbb{R}^{m \times r}$ which is generic. Let $\widetilde{C} \in \mathbb{R}^{m \times r}$ be such that* $\text{Supp}(\widetilde{C}) \subseteq \text{Supp}(C)$ *and the non-zero entries of $\widetilde{C}$ are the same as the corresponding non-zero entries of C. If $\widetilde{C}$ is full column rank, then C is also full column rank, almost surely.*

Proof: Since $\widetilde{C}$ is full column rank, there exists a $r \times r$ submatrix of $\widetilde{C}$, denoted by $\widetilde{C}_S$, with non-zero determinant, i.e., $\det(\widetilde{C}_S) \neq 0$. Let C_S denote the corresponding submatrix of C indexed by the same rows and columns as $\widetilde{C}_S$.

The determinant of C_S is a polynomial in the entries of C_S. Since $\widetilde{C}_S$ can be derived from C_S by keeping the corresponding non-zero entries, $\det(C_S)$ can be decomposed into two terms as

$$\det(C_S) = \det(\widetilde{C}_S) + f(C_S),$$

where the first term corresponds to the monomials for which all the variables (entries of C_S) are also in $\widetilde{C}_S$ and the second term corresponds to the monomials for which at least one variable is not in $\widetilde{C}_S$. The first term is non-zero as stated earlier. Since C is generic, the polynomial $f(C_S)$ is non-trivial and therefore its roots have Lebesgue measure zero. It implies that $\det(C_S) \neq 0$ with

Lebesgue measure one (almost surely), and hence, it is full (column) rank. Thus, C is also full column rank, almost surely. □

Finally, Theorem 5.1 is proved by combining the results of Theorem E.1 and Lemma 5.

Proof of Theorem 5.1: Since conditions 2 and 3 hold and A is generic, Lemma 5 can be applied which results that rank condition 6 is satisfied almost surely and expansion condition 7 also holds. Therefore, all the required conditions for Theorem E.1 are satisfied almost surely and this completes the proof. □

E.2 Proof of Random Identifiability Result (Theorem 5.2)

We provide detailed proof of the steps stated in the proof sketch of random result in Section 5.7.2.

E.2.1 Proof of Existence of Perfect n-gram Matching and Kruskal Results

Restatement of Theorem 5.3 *Consider a random bipartite graph $G(Y, X; E)$ with $|Y| = q$ nodes on the left side and $|X| = p$ nodes on the right side, and each node $i \in Y$ is randomly connected to d_i different nodes in X. Let $d_{\min} := \min_{i \in Y} d_i$. Assume that it satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ (condition 4) for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq \max\{1 + \beta \log p, \alpha \log p\}$ for some constants $\beta > \frac{n-1}{\log 1/c}, \alpha > \max\left\{2n^2\left(\beta \log \frac{1}{c} + 1\right), 2\beta n\right\}$ (lower bound in condition 5). Then, there exists a perfect (Y-saturating) n-gram matching in the random bipartite graph $G(Y, X; E)$, with probability at least $1 - \gamma_1 p^{-\beta'}$ for constants $\beta' > 0$ and $\gamma_1 > 0$, specified in (5.5) and (5.6).*

Proof of Theorem 5.3: Vertex sets X and Y are partitioned, described as follows (see Figure E.1). Define $J := c\frac{p}{n}$. Partition set X uniformly at random into n sets of (almost) equal size[7], denoted by $X'_l, l \in [n]$. Define sets $X_l := \cup_{i=1}^{l} X'_i, l \in [n]$. Furthermore, partition set Y uniformly at random, hierarchically as follows. First, partition into J sets, each with size at most $\left(c\frac{p}{n}\right)^{n-1}$,

[7]By almost, we mean the maximum difference in the size of partitions is 1 which is always possible.

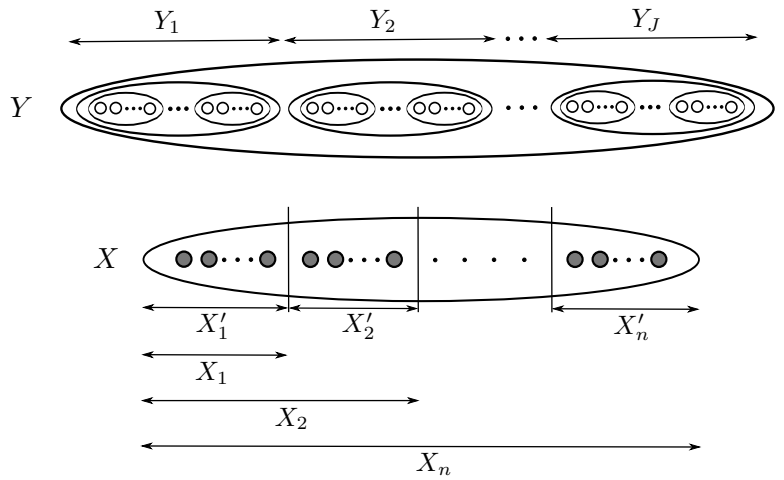


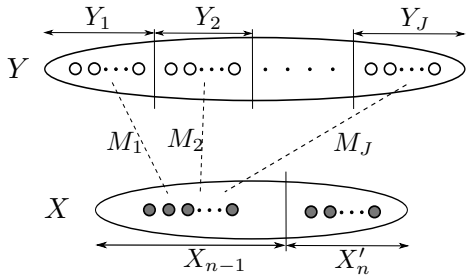
Figure E.1: Partitioning of sets Y and X, proposed in the proof of Theorem 5.3. Set X is randomly (uniform) partitioned into n sets of (almost) equal size, denoted by $X'_l, l \in [n]$. Set Y is also randomly partitioned in a recursive manner. In each step, it is partitioned to $J = c\frac{p}{n} = O(p)$ number of sets. These smaller sets are again partitioned, recursively. This partitioning process is performed until reaching sets with size $O(p)$. The first two steps are shown in this figure.

and denote them by $Y_i, i \in [J]$. Next, partition each of these new smaller sets Y_i further into J sets, each with size at most $\left(c\frac{p}{n}\right)^{n-2}$. Do it iteratively up to $n-1$ steps, where at the end, set Y is partitioned into sets with size at most $c\frac{p}{n}$. The first two steps are shown in Figure E.1.

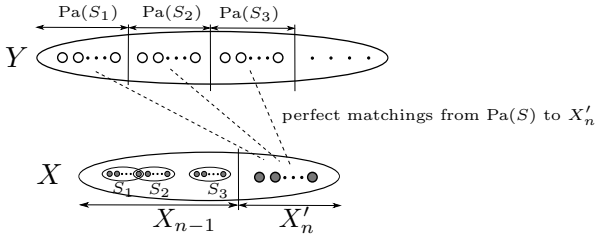
Proof by induction: The existence of perfect n-gram matching from set Y to set X is proved by an induction argument. Consider one of intermediate sets in the hierarchical partitioning of Y with size $O(p^l)$ and its further partitioning into $J := c\frac{p}{n}$ sets, each with size $O(p^{l-1})$, for any $l \in \{2, \ldots, n\}$. In the induction step, it is shown that if there exists a perfect $(l-1)$-gram matching from each of these subsets of Y with size $O(p^{l-1})$ to X_{l-1}, then there exists a perfect l-gram matching from the original set with size $O(p^l)$ to set X_l. Specifically, in the last induction step, it is shown that if there exists a perfect $(n-1)$-gram matching from each set $Y_l, l \in [J]$, to set X_{n-1}, then there exists a perfect n-gram matching from Y to $X_n = X$.

Base case of induction: The base case of induction argument holds as follows. By applying Lemma 8 and Lemma 7, there exists a perfect matching from each partition in Y with size at most $c\frac{p}{n} = O(p)$ to set X_1, **whp**.

Induction step: Consider J different bipartite graphs $G_i(Y_i, X_{n-1}; E_i), i \in [J]$, by considering sets Y_i and X_{n-1} and the corresponding subset of edges $E_i \subset E$ incident to them. See Figure E.2a. The



(a) Partitioning of sets Y and X proposed for the induction step.



(b) Partitioning of set Y through perfect $(n-1)$-gram matchings $M_i, i \in [J]$.

Figure E.2: Auxiliary figures for proof of induction step. (a) Partitioning of sets Y and X proposed in the proof, where set Y is partitioned to $J := c\frac{p}{n}$ partitions $Y_1, \ldots, Y_J$ with (almost) equal size, for some constant $c < 1$. In addition, set X is partitioned to two partitions X_{n-1} and X'_n with sizes $|X_{n-1}| = \frac{n-1}{n}p$ and $|X'_n| = \frac{p}{n}$. The perfect $(n-1)$-gram matchings $M_i, i \in [J]$, through bipartite graphs $G_i(Y_i, X_{n-1}; E_i), i \in [J]$, are also highlighted in the figure. (b) Set Y is partitioned to subsets $\mathrm{Pa}(S), S \in P_{n-1}(X_{n-1})$, which is generated through perfect $(n-1)$-gram matchings $M_i, i \in [J]$. S_1, S_2 and S_3 are three different sets in $P_{n-1}(X_{n-1})$ shown as samples. In addition, the perfect matchings from $\mathrm{Pa}(S), S \in P_{n-1}(X_{n-1})$, to X'_n, proposed in the proof, are also highlighted in the figure.

induction step is to show that if each of the corresponding J bipartite graphs $G_i(Y_i, X_{n-1}; E_i), i \in [J]$, has a perfect $(n-1)$-gram matching, then **whp**, the original bipartite graph $G(Y, X; E)$ has a perfect n-gram matching.

Let us denote the corresponding perfect $(n-1)$-gram matching of $G_i(Y_i, X_{n-1}; E_i)$ by M_i. Furthermore, the set of all subsets of X_{n-1} with cardinality $n-1$ are denoted by $P_{n-1}(X_{n-1})$, i.e., $P_{n-1}(X_{n-1})$ includes the sets with $(n-1)$ elements in the power set[8] of X_{n-1}. For each set $S \in P_{n-1}(X_{n-1})$, take the set of all nodes in Y which are connected to all members of S according to the union of matchings $\cup_{i=1}^{J} M_i$. Call this set the parents of S, denoted by $\mathrm{Pa}(S)$. According to the definition of perfect $(n-1)$-gram matching, there is at most one node in each set Y_i which is connected to all members of S through the matching M_i and therefore, $|\mathrm{Pa}(S)| \leq J = c\frac{p}{n}$. In addition, note that sets $\mathrm{Pa}(S)$ impose a partitioning on set Y, i.e., each node $j \in Y$ is exactly included in one set $\mathrm{Pa}(S)$ for some $S \in P_{n-1}(X_{n-1})$. This is because of the perfect $(n-1)$-gram matchings considered for sets $Y_i, i \in [J]$.

Now, a perfect n-gram matching for the original bipartite graph is constructed as follows. For any $S \in P_{n-1}(X_{n-1})$, consider the set of parents $\mathrm{Pa}(S)$. Create the bipartite graph $G_S(\mathrm{Pa}(S), X'_n; E_S)$,

[8]The power set of any set S is the set of all subsets of S.

where $E_S \subset E$ is the subset of edges incident to partitions $\mathrm{Pa}(S) \subset Y$ and $X'_n \subset X$. Denote by d_S the minimum degree of nodes in set $\mathrm{Pa}(S)$ in the bipartite graph $G_S(\mathrm{Pa}(S), X'_n; E_S)$. Applying Lemma 8, we have

$$\begin{aligned}\Pr[d_S \geq 1 + \beta \log(p/n)] &\geq 1 - J \exp\left(-\frac{2}{n^2}\frac{(d_{\min} - \beta n \log(p/n))^2}{d_{\min}}\right) \\ &\geq 1 - \frac{c}{n} p^{-\beta \log 1/c} = 1 - O(p^{-\beta \log 1/c}),\end{aligned} \tag{E.14}$$

where $\beta \log 1/c > n-1$, and the last inequality is concluded from the degree bound $d_{\min} \geq \alpha \log p$. Furthermore, we have $|\mathrm{Pa}(S)| \leq c\frac{p}{n} = c|X'_n|$. Now, we can apply Lemma 7 concluding that there exists a perfect matching from $\mathrm{Pa}(S)$ to X'_n within the bipartite graph $G_S(\mathrm{Pa}(S), X'_n; E_S)$, with probability at least $1 - O(p^{-\beta \log 1/c})$. Refer to Figure E.2b for a schematic picture. The edges of this perfect matching are combined with the corresponding edges of the existing perfect $(n-1)$-gram matchings $M_i, i \in [J]$, to provide n incident edges to each node $i \in \mathrm{Pa}(S)$. It is easy to see that this provides a perfect n-gram matching from $\mathrm{Pa}(S)$ to X.

We perform the same steps for all sets $S \in P_{n-1}(X_{n-1})$ to obtain a perfect n-gram matching from any $\mathrm{Pa}(S), S \in P_{n-1}(X_{n-1})$, to X. Finally, according to this construction, the union of all of these matchings is a perfect n-gram matching from $\cup_{S \in P_{n-1}(X_{n-1})} \mathrm{Pa}(S) = Y$ to X. This finishes the proof of induction step. Note that here we analyzed the last induction step where the existence of perfect n-gram matching is concluded from the existence of corresponding perfect $(n-1)$-gram matchings. The earlier induction steps, where the existence of perfect l-gram matching is concluded from the existence of corresponding perfect $(l-1)$-gram matchings for any $l \in \{2, \ldots, n\}$, can be similarly proven.

Probability rate: We now provide the probability rate of the above events. Let $N_l^{(\mathrm{hp})}, l \in [n]$, denote the total number of times that perfect matching result of Lemma 7 is used in step l in order to ensure that there exists a perfect l-gram matching from corresponding partitions of Y to set X_l, **whp**. Let $N^{(\mathrm{hp})} = \sum_{l \in [n]} N_l^{(\mathrm{hp})}$. As earlier, let $P_{l-1}\big(X_{l-1}\big)$ denote the set of all subsets of X_{l-1} with cardinality $l-1$. We have

$$\big|P_{l-1}(X_{l-1})\big| = \binom{\big|X_{l-1}\big|}{l-1} = \binom{\frac{l-1}{n}p}{l-1}, \quad l \in \{2, \ldots, n\}.$$

According to the construction method of l-gram matching from $(l-1)$-gram matchings, proposed in the induction step, $\left|P_{l-1}\left(X_{l-1}\right)\right|$ is the number of times Lemma 7 is used in order to ensure that there exists a perfect l-gram matching for each partition on the Y side. Since at most J^{n-l} number of such l-gram matchings are proposed in step l, the number $N_l^{(\mathrm{hp})}$ can be bounded as

$$N_l^{(\mathrm{hp})} \leq J^{n-l}\left|P_{l-1}\left(X_{l-1}\right)\right| = J^{n-l}\binom{\frac{l-1}{n}p}{l-1}, \quad l \in \{2,\ldots,n\}. \tag{E.15}$$

Since in the first step, $N_1^{(\mathrm{hp})} = J^{n-1}$ number of perfect matchings needs to exist in the above discussion, we have

$$\begin{aligned}
N^{(\mathrm{hp})} &= J^{n-1} + \sum_{l=2}^{n} N_l^{(\mathrm{hp})} \\
&\leq J^{n-1} + \sum_{l=2}^{n} J^{n-l}\binom{\frac{l-1}{n}p}{l-1} \\
&\leq \left(c\frac{p}{n}\right)^{n-1} + \sum_{l=2}^{n}\left(c\frac{p}{n}\right)^{n-l}\left(e\frac{p}{n}\right)^{l-1} \\
&\leq n\left(e\frac{p}{n}\right)^{n-1} = O(p^{n-1}),
\end{aligned}$$

where inequality (E.15) is used in the first inequality and $J := c\frac{p}{n}$ and inequality $\binom{n}{k} \leq \left(e\frac{n}{k}\right)^k$ are exploited in the second inequality.

Since the result of Lemma 7 holds with probability at least $1 - O(p^{-\beta \log 1/c})$ and it is assumed that $\beta \log 1/c > n-1$, by applying union bound, we have the existence of perfect n-gram matching with probability at least $1 - O(p^{-\beta'})$, for $\beta' = \beta \log \frac{1}{c} - (n-1) > 0$.

Furthermore, note that the degree concentration bound in (E.14) is also used $O(p^{n-1})$ times. Since the bound in (E.14) holds with probability at least $1 - O(p^{-\beta \log 1/c})$ and it is assumed that $\beta \log 1/c > n-1$, this also reduces to the same probability rate.

The coefficient of the above polynomial probability rate is also explicitly computed, saying that the perfect n-gram matching exists with probability at least $1 - \gamma_1 p^{-\beta'}$, with

$$\gamma_1 = e^{n-1}\left(\frac{c}{n^{n-1}} + \frac{e^2}{1-\delta_1}n^{\beta'+1}\right),$$

where δ_1 is a constant satisfying $e^2\left(\frac{p}{n}\right)^{-\beta \log 1/c} < \delta_1 < 1$. □

Proof of Theorem 5.4: Let $G(Y, X; A)$ denote the corresponding bipartite graph to matrix A where node sets $Y = [q]$ and $X = [p]$ index the columns and rows of A respectively. Therefore, $|Y| = q$ and $|X| = p$. Fix some $S \subseteq Y$ such that $|S| \leq p$. Then

$$\begin{aligned}
\Pr(|N(S)| \leq |S|) &\leq \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \Pr(N(S) \subseteq T) \\
&= \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \prod_{i \in S} \binom{|S|}{d_i} \Big/ \binom{p}{d_i} \\
&\leq \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \prod_{i \in S} \left(\frac{|S|}{p}\right)^{d_i} \\
&\leq \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \prod_{i \in S} \left(\frac{|S|}{p}\right)^{d_{\min}} \\
&= \binom{p}{|S|} \left(\frac{|S|}{p}\right)^{d_{\min}|S|}, \qquad \text{(E.16)}
\end{aligned}$$

where the bound $\binom{|S|}{d_i} \Big/ \binom{p}{d_i} \leq \left(\frac{|S|}{p}\right)^{d_i}$ is used in the second inequality, and the last inequality is concluded from the fact that $\frac{|S|}{p} \leq 1$.

Let $\mathcal{E}$ denote the event that for any subset $S \subseteq Y$ with $|S| \leq r$, we have $|N(S)| \geq |S|$, i.e.,

$$\mathcal{E} := \text{“}\forall S \subseteq Y \wedge 1 \leq |S| \leq r : |N(S)| \geq |S|\text{”}.$$

Then, by the union bound and inequality (E.16), we have

$$\begin{aligned}
\Pr(\mathcal{E}^c) = \Pr(\exists S \subseteq Y \,\text{s.t.}\, 1 \leq |S| \leq r \wedge |N(S)| < |S|) &\leq \sum_{s=1}^{r} \binom{q}{s} \binom{p}{s} \left(\frac{s}{p}\right)^{d_{\min}s} \\
&\leq \sum_{s=1}^{r} \left(e\frac{q}{s}\right)^{s} \left(e\frac{p}{s}\right)^{s} \left(\frac{s}{p}\right)^{d_{\min}s} \\
&\leq \sum_{s=1}^{r} \left(\frac{e^2 q r^{d_{\min}-2}}{p^{d_{\min}-1}}\right)^{s},
\end{aligned}$$

where the bound $\binom{n}{k} \leq \left(e\frac{n}{k}\right)^k$ is used in the second inequality. For $r = cp$, the above inequality reduces to

$$\begin{aligned}
\Pr(\mathcal{E}^c) &\leq \sum_{s=1}^{r} \left(e^2 c^{d_{\min}-2} \frac{q}{p} \right)^s \\
&\leq \sum_{s=1}^{r} \left(e^2 c' c^{d_{\min}-1} p^{n-1} \right)^s \\
&\leq \sum_{s=1}^{r} \left(e^2 c' c^{\beta \log p} p^{n-1} \right)^s \\
&= \sum_{s=1}^{r} \left(e^2 c' p^{n-1-\beta \log 1/c} \right)^s \\
&\leq \frac{e^2 c'}{p^{\beta'} - e^2 c'} = O(p^{-\beta'}), \quad \text{for } \beta' = \beta \log \frac{1}{c} - (n-1) > 0,
\end{aligned}$$

where the size condition assumed in the theorem is used in the second inequality with $c' := \frac{1}{c}\left(\frac{c}{n}\right)^n$, and the degree condition is exploited in the third inequality. The last inequality is concluded from the geometric series sum formula for large enough p.

Then, Lemma 9 can be applied concluding that $\text{krank}(A) \geq r = cp$, with probability at least $1 - \gamma_2 p^{-\beta'}$ for constants $\beta' = \beta \log \frac{1}{c} - (n-1) > 0$ and $\gamma_2 > 0$ as

$$\gamma_2 = \frac{c^{n-1} e^2}{n^n (1 - \delta_2)},$$

where δ_2 is a constant satisfying $c' e^2 p^{-\beta'} < \delta_2 < 1$. □

Proof of Remark 25: Consider a random bipartite graph $G(Y, X; E)$ where for each node $i \in X$:

1. Neighbors $N(i) \subseteq X$ are picked uniformly at random among all size d subsets of X.
2. Matching $M(i) \subseteq N(i)$ is picked uniformly at random among all size n subsets of $N(i)$.

Note that as long as $n \leq d$, the distribution of $M(i)$ is uniform over all size n subsets of X.

Fix some pair $i, i' \in Y$. Then

$$\Pr(M(i) = M(i')) = \binom{|X|}{n}^{-1}.$$

By the union bound,

$$\Pr\Big(\exists i, i' \in Y, i \neq i' \,\mathrm{s.t.}\, M(i) = M(i')\Big) \leq \binom{|Y|}{2}\binom{|X|}{n}^{-1},$$

which is $\Theta(|Y|^2/|X|^n)$ when n is constant. Therefore, if $d \geq n$ and the size constraint $|Y| = O(|X|^s)$ for some $s < \frac{n}{2}$ is satisfied, then **whp**, there is no pair of nodes in set Y with the same random n-gram matching. This concludes that the random bipartite graph has a perfect n-gram matching **whp**, under these size and degree conditions.

□

E.2.2 Auxiliary Lemmata

Lemma 7 (Existence of perfect matching for random bipartite graphs)**.** *Consider a random bipartite graph $G(W, Z; E)$ with $|W| = w$ nodes on the left side and $|Z| = z$ on the right side, and each node $i \in W$ is randomly connected to d_i different nodes in set Z. Let $d_w := \min_{i \in W} d_i$. Assume that it satisfies the size condition $w \leq cz$ for some constant $0 < c < 1$ and the degree condition $d_w \geq 1 + \beta \log z$ for some constant $\beta > 0$. Then, there exists a perfect matching in the random bipartite graph $G(W, Z; E)$ with probability at least $1 - O(z^{-\beta \log 1/c})$ where $\beta \log \frac{1}{c} > 0$.*

Proof: From Hall's theorem (Theorem E.2), the existence of perfect matching for a bipartite graph is equivalent to occurrence of the following event

$$\widetilde{\mathcal{E}} := \text{“}\forall S \subseteq W : |N(S)| \geq |S|\text{”}.$$

Similar to the analysis in the proof of Theorem 5.4, applying the union bound we have

$$\begin{aligned}\Pr\big(\widetilde{\mathcal{E}}^c\big) = \Pr(\exists S \subseteq W \,\text{s.t.}\, |N(S)| < |S|) &\le \sum_{s=1}^{w} \binom{w}{s}\binom{z}{s}\left(\frac{s}{z}\right)^{d_w s} \\ &\le \sum_{s=1}^{w}\left(e\frac{w}{s}\right)^{s}\left(e\frac{z}{s}\right)^{s}\left(\frac{s}{z}\right)^{d_w s} \\ &\le \sum_{s=1}^{w}\left(\frac{e^2 w^{d_w-1}}{z^{d_w-1}}\right)^{s} \\ &\le \sum_{s=1}^{w}\Big(e^2 c^{d_w-1}\Big)^{s},\end{aligned}$$

where the bound $\binom{n}{k} \le \big(e\frac{n}{k}\big)^k$ is used in the second inequality. From the assumed lower bound on the degree d_w and the fact that $0 < c < 1$, we have

$$\Pr\big(\widetilde{\mathcal{E}}^c\big) \le \sum_{s=1}^{w}\Big(e^2 c^{\beta \log z}\Big)^{s} = \sum_{s=1}^{w}\Big(e^2 z^{\beta \log c}\Big)^{s} \le \frac{e^2}{z^{\beta \log \frac{1}{c}} - e^2} \le \frac{e^2}{1-\delta_1} z^{-\beta \log 1/c},$$

where the second inequality is concluded from the geometric series sum formula for large enough z, and δ_1 is a constant satisfying $e^2 z^{-\beta \log 1/c} < \delta_1 < 1$. □

Lemma 8 (Degree concentration bound)**.** *Consider a random bipartite graph $G(Y, X; E)$ with $|Y| = q$ and $|X| = p$, where each node $i \in Y$ is randomly connected to d_i different nodes in set X. Let $Y' \subset Y$ be any subset[9] of nodes in Y with size $|Y'| = q'$ and $X' \subset X$ be a random (uniformly chosen) subset of nodes in X with size $|X'| = p'$. Create the new bipartite graph $G(Y', X'; E')$ where edge set $E' \subset E$ is the subset of edges in E incident to Y' and X'. Denote the degree of each node $i \in Y'$ within this new bipartite graph by d'_i. Let $d_{\min} := \min_{i\in Y} d_i$ and $d'_{\min} := \min_{i\in Y'} d'_i$. Then, if $d_{\min} > r\frac{p}{p'}$ for a non-negative integer r, we have*

$$\Pr[d'_{\min} \ge r+1] \ge 1 - q' \exp\left(-2(p'/p)^2 \frac{(d_{\min} - (p/p')r)^2}{d_{\min}}\right).$$

Proof: For any $i \in Y'$, we have

$$\Pr[d'_i \le r] = \sum_{j=0}^{r}\binom{p'}{j}\binom{p-p'}{d_i - j}\Big/\binom{p}{d_i},$$

[9]Note that Y' need not to be uniformly chosen and the result is valid for any subset of nodes $Y' \subset Y$.

where the inner term of summation is a hypergeometric distribution with parameters p (population size), p' (number of success states in the population), d_i (number of draws) and j is the hypergeometric random variable denoting number of successes. The following tail bound for the hypergeometric distribution is provided [56, 146]

$$\Pr[d'_i \leq r] \leq \exp(-2t_i^2 d_i),$$

for $t_i > 0$ given by $r = \big(\frac{p'}{p} - t_i\big)d_i$. Note that assumption $d_{\min} > \frac{p}{p'}r$ in the lemma is equivalent to having $t_i > 0, i \in Y$. Considering the minimum degree, for any $i \in Y'$, we have

$$\Pr[d'_i \leq r] \leq \exp(-2t^2 d_{\min}),$$

for $t > 0$ given by $r = \big(\frac{p'}{p} - t\big)d_{\min}$. Substituting t from this equation gives the following bound

$$\Pr[d'_i \leq r] \leq \exp\bigg(-2(p'/p)^2 \frac{(d_{\min} - (p/p')r)^2}{d_{\min}}\bigg). \tag{E.17}$$

Finally, applying the union bound, we can prove the result as follows

$$\begin{aligned}
\Pr[d'_{\min} \geq r+1] =& \Pr[\cap_{i=1}^{q'}\{d'_i \geq r+1\}] \\
\geq& 1 - \sum_{i=1}^{q'} \Pr[d'_i \leq r] \\
\geq& 1 - \sum_{i=1}^{q'} \exp\bigg(-2(p'/p)^2 \frac{(d_{\min} - (p/p')r)^2}{d_{\min}}\bigg) \\
=& 1 - q' \exp\bigg(-2(p'/p)^2 \frac{(d_{\min} - (p/p')r)^2}{d_{\min}}\bigg),
\end{aligned}$$

where the union bound is applied in the first inequality and the second inequality is concluded from (E.17). □

A lower bound on the Kruskal rank of matrix A based on a sufficient relaxed expansion property on A is provided in the following lemma which might have independent interest.

Lemma 9. *If A is generic and the bipartite graph $G(Y, X; A)$ satisfies the relaxed*[10] *expansion property $|N(S)| \geq |S|$ for any subset $S \subseteq Y$ with $|S| \leq r$, then* $\text{krank}(A) \geq r$*, almost surely.*

Before proposing the proof, we state the marriage or Hall's theorem which gives an equivalent condition for having a perfect matching in a bipartite graph.

Theorem E.2 (Hall's theorem, [83])**.** *A bipartite graph $G(Y, X; E)$ has Y-saturating matching if and only if for every subset $S \subseteq Y$, the size of the neighbors of S is at least as large as S, i.e., $|N(S)| \geq |S|$.*

Proof of Lemma 9: Denote the submatrix $A_{N(S),S}$ by $\widetilde{A}_S$, i.e., $\widetilde{A}_S := A_{N(S),S}$. Exploiting marriage or Hall's theorem, it is concluded that the bipartite graph $G(S, N(S); \widetilde{A}_S)$ has a perfect matching M_S for any subset $S \subseteq Y$ such that $|S| \leq r$. Denote by $\widetilde{A}_{M_S}$ the corresponding matrix to this perfect matching edge set M_S, i.e., $\widetilde{A}_{M_S}$ keeps the non-zero entries of $\widetilde{A}_S$ on edge set M_S and everywhere else, it is zero. Note that the support of $\widetilde{A}_{M_S}$ is within the support of $\widetilde{A}_S$. According to the definition of perfect matching, the matrix $\widetilde{A}_{M_S}$ is full column rank. From Lemma 6, it is concluded that $\widetilde{A}_S$ is also full column rank almost surely. This is true for any $\widetilde{A}_S$ with $S \subseteq Y$ and $|S| \leq r$, which directly results that $\text{krank}(A) \geq r$, almost surely. □

Finally, Theorem 5.2 is proved by exploiting the random results on the existence of perfect n-gram matching and Kruskal rank, provided in Theorems 5.3 and 5.4.

Proof of Theorem 5.2: We claim that if random conditions 4 and 5 are satisfied, then deterministic conditions 2 and 3 hold **whp**. Then Theorem 5.1 can be applied and the proof is done. From size and degree conditions, Theorem 5.3 can be applied, which implies that the perfect n-gram matching condition 2 is satisfied with probability at least $1 - \gamma_1 p^{-\beta'}$ for $\beta' = \beta \log \frac{1}{c} - (n-1) > 0$. The conditions required for Theorem 5.4 also hold and by applying this theorem we have the bound $\text{krank}(A) \geq cp$, with probability at least$1-\gamma_2 p^{-\beta'}$. Combining this inequality with the upper bound on degree d in condition 5, we conclude that krank condition 3 is also satisfied **whp**. Hence, all the

[10]There is no $d_{\max}$ term in contrast to the expansion property proposed in condition 7.

conditions required for Theorem 5.1 are satisfied with probability at least $1 - \gamma p^{-\beta'}$, where

$$\gamma = \gamma_1 + \gamma_2 = e^{n-1}\Big(\frac{c}{n^{n-1}} + \frac{e^2}{1-\delta_1} n^{\beta'+1}\Big) + \frac{c^{n-1}e^2}{n^n(1-\delta_2)},$$

and this completes the proof. □

Finally, Corollary 5.2 can be also proved by showing that the size and degree conditions satisfy the full column rank condition required in Corollary 5.1. This is proved in Lemma 7.

E.3 Relationship to CP Decomposition Uniqueness Results

In this section, we provide a more detailed comparison with some uniqueness results of overcomplete CP decomposition. Here, the following CP decomposition for the third order tensor $T \in \mathbb{R}^{p\times s\times q}$ is considered,

$$T = \sum_{i=1}^{r} a_i \circ b_i \circ c_i, \tag{E.18}$$

where $A = [a_1|\dots|a_r] \in \mathbb{R}^{p\times r}, B = [b_1|\dots|b_r] \in \mathbb{R}^{s\times r}$ and $C = [c_1|\dots|c_r] \in \mathbb{R}^{q\times r}$.
The most important and general uniqueness result of CP, called Kruskal's condition, is provided in [112], where it is guaranteed that the above CP decomposition is unique if

$$\text{krank}(A) + \text{krank}(B) + \text{krank}(C) \geq 2r + 2.$$

Since then, several works have analyzed the uniqueness of CP decomposition. One set of works assume that one of the components, say C, is full column rank [116, 103]. It is shown in [116], for generic (fully dense) components A, B and C, if $r \leq q$ and $r(r-1) \leq p(p-1)s(s-1)/2$, then the CP decomposition in (E.18) is generically unique.
Now, we demonstrate how this CP uniqueness result can be adapted to our setting. First, consider

the matrix $M \in \mathbb{R}^{ps \times q}$ which is obtained by stacking the entries of T as

$$M_{(i-1)s+j,k} = T_{ijk}.$$

Then, we have

$$M = (A \odot B)C^\top. \tag{E.19}$$

On the other hand, for the 2-persistent topic model with 4 words ($n = 2, m = 2$), the moment can be written as

$$M_4^{(2)}(x) = (A \odot A)\mathbb{E}\big[hh^\top\big](A \odot A)^\top,$$

for $A \in \mathbb{R}^{p \times q}$. The following matrix has the same column span of $M_4^{(2)}(x)$,

$$M' = (A \odot A)C'^\top,$$

for some full rank matrix $C' \in \mathbb{R}^{q \times q}$. Our random identifiability result in Theorem 5.2 provides the uniqueness of A and C', given M', under the size condition $q \leq \binom{p}{2}^2$ and the additional degree condition 5. Note that as discussed in the previous section, this identifiability argument is the same as the unique decomposition of the corresponding tensor.

Thus, in equation (E.19), by setting $A = B$ and a full rank square matrix C, we obtain the 2-persistent topic model, under consideration in this work. Thus, the identifiability results of [116] are applicable to our setting, if we assume generic (i.e. fully dense) matrix A. However, we incorporate a sparse matrix A, and therefore, require different techniques to provide identifiability results. We note that the size bound specified in [116] is comparable to the size bound derived in this work (for random structured matrices), but we have additional degree considerations for identifiability. Analyzing the regime where the uniqueness conditions of [116] are satisfied under sparsity constraints is an interesting question for future investigation.