THE UNIVERSITY OF CHICAGO

FAST NUMERICAL LINEAR ALGEBRA FOR GENERATIVE MODELING

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

COMMITTEE ON COMPUTATIONAL AND APPLIED MATHEMATICS

BY
YIFAN PENG

CHICAGO, ILLINOIS
JUNE 2026

To everyone who has been part of my journey

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Yuehaw Khoo. He introduced me to the fascinating area of generative modeling through the perspective of numerical linear algebra. Under his guidance, I learned to cultivate numerical intuition for the phenomena observed in the experiments and to analyze the underlying mechanisms with clarity and rigor. It has been both a privilege and an honor to work with him. Beyond technical skills, I have benefited greatly from his passion for scientific discovery and his thoughtful, critical approach to problem solving. I am also sincerely thankful for his unwavering support and invaluable advice regarding my future career. His mentorship has been a constant source of inspiration and has profoundly shaped both my academic development and personal growth.

I would also like to express my sincere appreciation to my co-advisor, Professor Daren Wang. He helped me develop strong statistical intuition, which complements my numerical perspective and broadens my approach to research. His encouragement and steady support have enabled me to overcome challenges and continue striving for improvement throughout my PhD.

I am grateful to my committee members, Professor Mihai Anitescu and Professor Lek-Heng Lim, for their time, dedication, and insightful feedback. Their thoughtful suggestions encouraged deeper reflection and led to a more comprehensive exploration of my research topics. I also sincerely thank them for their courses, Matrix Computation and Numerical Optimization, during my first year, which provided me with a strong foundational background for my later research. Furthermore, I would like to thank Professor Mathias Oster for the rewarding collaboration on an engaging project. His insightful suggestions were instrumental in shaping the project and bringing it to completion smoothly. I am also deeply grateful for his support during my visit to Germany, which became one of the most peaceful and memorable periods of my Ph.D. journey.

I am also grateful to E. Miles Stoudenmire for his guidance and kindness during the early stages of my doctoral studies. My sincere thanks go to Yian Chen for his advice on planning my Ph.D. journey and for sharing valuable perspectives on future career paths. I would also like to express my appreciation to Siyao Yang for our collaboration over the past three years and for the many insightful discussions we have had on both research and life.

Finally, I would like to extend my heartfelt thanks to the friends I have met throughout my PhD journey. In a rapidly changing world, I regard each relationship as a form of serendipity, and I deeply value the kindness and encouragement I have received along the way. I am also grateful for my beloved cat whose companionship has brought warmth to my daily life. I would like to express my deepest appreciation to my parents for their unconditional trust and unwavering support. Their encouragement has helped me overcome setbacks and face challenges with renewed strength whenever I felt discouraged. All of this care and encouragement has made my journey more iridescent and meaningful.

ABSTRACT

Generative modeling aims to learn the underlying probability distribution of observed data and to generate new samples that resemble the training data. In recent years, deep neural networks have achieved remarkable success in high-dimensional data generation tasks, particularly in image and audio generation. Despite the success, many modern architectures rely on overparametrized neural networks, which can be computationally expensive and difficult to interpret and reveal the intrinsic structure of the underlying distribution.

In this thesis, we investigate an alternative framework for generative modeling based on techniques from numerical linear algebra. Chapters 2–4 focus on density estimation and develop a series of methods based on tensor network representations. Tensor networks provide structured and interpretable representations for high-dimensional functions, offering significant advantages in terms of computational efficiency and scalability. In Chapter 2, we introduce an efficient variance-reduction scheme that improves upon the standard higher-order singular value decomposition, enabling its application to moderately high-dimensional problems. Chapter 3 presents a tensor-train-based density estimation framework, where a novel convolution–deconvolution strategy is proposed to mitigate the exponential variance growth associated with empirical density estimation in high dimensions. In Chapter 4, we extend these ideas to hierarchical tensor-train representations. The ultimate goal of the series is to achieve linear computational complexity in the dimensionality and maintain high accuracy in many applications.

Chapters 5 and 6 focus on diffusion-based approaches. In Chapter 5, we develop a numerical linear algebra framework for diffusion models that enables optimization-free score estimation. Chapter 6 further extends this approach by incorporating sparse regression techniques to recover sparse structures. Extensive numerical experiments demonstrate the effectiveness of the proposed methods, and theoretical analysis establishes error bounds that characterize their dependence on the dimensionality.

CHAPTER 1

INTRODUCTION

Generative modeling aims to learn the underlying probability distribution of observed data and to generate new samples that resemble the training data. Formally, given a dataset $\{x^{(i)}\}_{i=1}^{N}$ independently and identically drawn from an unknown distribution p^*, the goal is to construct a model $\tilde{p}$ such that $\tilde{p} \approx p^*$. Once such a model is obtained, it can be used for sampling as well as a variety of downstream tasks. This problem has broad applications across multiple disciplines, including science [43, 33], engineering [29], economics [243], and modern machine learning [23, 116, 220, 105].

Methods for generative modeling can be broadly categorized into two classes. The first class consists of approaches that aim to explicitly estimate the probability density function, commonly referred to as density estimation in the statistical literature. Classical methods such as histograms [193, 195] and kernel density estimators [172, 45] are well known for their numerical robustness and statistical stability in low-dimensional settings, but they often suffer from the curse of dimensionality as the dimension increases. Mixture models [49, 63] provide a more flexible framework for modeling complex distributions; however, they may still be limited in capturing highly nonparametric structures. To address these challenges, adaptive methods [135, 137] have been developed for high-dimensional settings. In recent years, deep learning has emerged as a powerful tool for approximating high-dimensional probability distributions from data, particularly in applications such as image and signal processing. Representative approaches include energy-based models [127, 91, 157], normalizing flows [52, 184, 53], and autoregressive models [68, 217, 170, 99]. Energy-based models assign an energy value to each data configuration and define probability through relative comparisons of these energies, where lower energy corresponds to higher likelihood; they are typically trained by shaping the energy landscape using sampling-based methods. Normalizing flows construct invertible mappings between a simple base distribution and the target

distribution, enabling exact likelihood evaluation and efficient sampling. Autoregressive models factorize high-dimensional distributions into a sequence of conditional distributions, allowing flexible modeling while maintaining tractable likelihood computation.

Another important class of generative models is based on implicit distributions, where sampling is straightforward while likelihood evaluation is intractable. Generative adversarial networks (GANs) [70, 10, 78, 23, 105] formulate learning as a minimax game between a generator and a discriminator, and have demonstrated remarkable success in producing high-quality samples in complex domains such as images and audio. Variational autoencoders (VAEs) [117, 186, 90, 118] introduce latent variable models and optimize a variational lower bound of the likelihood, providing a principled and scalable framework for probabilistic inference and representation learning. More recently, score-based generative models and diffusion models have achieved state-of-the-art performance in high-dimensional data generation tasks [203, 92, 206, 158, 51]. These approaches rely on estimating the score function $\nabla_x \log p(x)$ of the data distribution and generate samples by simulating stochastic differential equations. Research on diffusion models has primarily centered around two dominant formulations. The first approach is based on forward SDE simulation [92, 200, 205, 202], where a stochastic differential equation (typically an Ornstein–Uhlenbeck process) is used to gradually diffuse data into noise. The score function is then estimated by training a neural network. The second formulation is based on flow matching [134, 140, 138], which directly learns a time-dependent vector field that couples the target and base distributions, rather than estimating score functions through a forward stochastic process. Extensions of this approach include methods based on stochastic interpolants [7, 5, 6], which interpolate between distributions using latent variables. For an overview of these methods, see [238, 28, 145].

Despite their empirical success, many modern generative models rely on overparameterized neural networks, which can be computationally expensive and difficult to interpret. This has motivated the exploration of alternative approaches based on structured repre-

sentations, such as low-rank tensor decompositions and sparse function expansions, which aim to capture the intrinsic low-dimensional structure of high-dimensional distributions. These perspectives provide opportunities to integrate tools from numerical linear algebra and high-dimensional statistics into generative modeling, enabling scalable, interpretable, and theoretically grounded methods.

1.1 Organization

In this thesis, we develop several methods to address the aforementioned questions from the perspective of numerical linear algebra. Chapters 2–4 focus on density estimation, corresponding to the first class of generative models, and present a sequence of approaches based on tensor network structures for high-dimensional problems. In Chapter 2 (work [111]), we introduce an efficient variance-reduction scheme that improves upon the standard higher-order singular value decomposition, thereby extending its applicability to moderately high-dimensional settings. Chapter 3 (work [176]) presents a tensor-train-based framework for density estimation, in which a novel convolution–deconvolution strategy is proposed to mitigate the exponential variance growth arising in empirical density estimation in high dimensions. In Chapter 4 (work [175]), we further extend these ideas to hierarchical tensor-train representations. The final objective of these chapters is to develop methods whose computational complexity scales linearly with the dimensionality while maintaining strong performance across a range of numerical experiments.

Chapters 5–6 turn to diffusion models, which belong to the second class of generative models. In Chapter 5 (work [110]), we develop a numerical linear algebra framework for diffusion models that enables optimization-free score estimation. Chapter 6 further extends this framework by incorporating sparse regression techniques to recover sparse structures efficiently. The goal of these chapters is to characterize the dependence of the estimation error on the dimensionality through both theoretical analysis and numerical experiments.

1.2 Related Work

Here we summarize several numerical linear algebra techniques that we use in this thesis.

Tensor Network: Tensor network representations have emerged as powerful tools for the analysis and computation of high-dimensional problems, where classical methods suffer from the curse of dimensionality. By exploiting low-rank structures and separability in multi-dimensional data, tensor networks provide compact representations that significantly reduce storage and computational complexity. These methods have been widely applied in scientific computing, signal processing, machine learning, and quantum physics [119, 73, 112, 41].

One of the simplest and most classical tensor decompositions is the canonical polyadic (CP) decomposition, which represents a tensor as a sum of rank-one components [27, 88, 123, 124, 42]. The CP format is highly interpretable and memory-efficient when the rank is small. However, it can suffer from numerical instability and issues of ill-posedness, particularly in high-dimensional or noisy settings.

Another widely used representation is the Tucker decomposition, which expresses a tensor as a core tensor multiplied by factor matrices along each mode [216, 46, 221, 47]. The Tucker format can be viewed as a higher-order generalization of the matrix singular value decomposition and provides a flexible framework for dimensionality reduction and feature extraction. Despite its flexibility, the storage cost of the Tucker format grows exponentially with the tensor order, which limits its applicability in very high-dimensional problems.

To address this limitation, the tensor-train (TT) decomposition, also known as the matrix product state (MPS) ([233, 177]) in the physics literature, was introduced as a scalable tensor network format for high-dimensional problems [166, 167, 168, 165]. In the TT representation, a high-dimensional tensor is factorized into a sequence of low-dimensional core tensors connected in a chain structure. This representation enables both storage and computational costs to scale linearly with the dimension, making it particularly suitable for high-dimensional approximation and learning tasks.

Beyond the formats mentioned above, a variety of more advanced tensor network structures have been developed to further improve flexibility and scalability in high-dimensional settings. Hierarchical Tucker (HT) decompositions organize tensor representations in a tree-based structure, enabling efficient approximation with controlled ranks [81, 72]. More general tensor network architectures, such as tree tensor networks and projected entangled pair states (PEPS), originate from quantum many-body physics and provide powerful representations for capturing complex multi-way interactions [164, 222]. Tensor ring (TR) decompositions extend the tensor-train format by forming a cyclic structure, which can enhance representation flexibility while maintaining favorable computational complexity [244, 228, 109]. These advanced tensor network formats further expand the applicability of low-rank tensor methods and provide a rich framework for modeling and computation in high-dimensional problems.

Randomized Linear Algebra: A notable advancement in numerical linear algebra is the emergence of randomized low-rank approximation algorithms. These algorithms excel in reducing time and space complexity substantially without sacrificing too much numerical accuracy. Seminal contributions to this area are outlined in works such as [133] and [82]. Additionally, review papers like [235], [59], [149], and [150] have provided comprehensive summaries of these randomized approaches, along with their theoretical stability guarantees. Randomized low-rank approximation algorithms typically start by estimating the range of a large low-rank matrix $A \in \mathbb{R}^{n\times n}$ by forming a reduced-size sketch. This is achieved by right multiplying A with a random matrix $S \in \mathbb{R}^{n\times k}$, where $k \ll n$. The random matrix S is selected to ensure that the range of AS remains a close approximation of the range of A, even when the column size of AS is significantly reduced from A. As such, the random matrix S is referred to as randomized linear embedding or sketching matrix by [215] and [156]. The sketching approach reduces the cost in singular value decomposition from $\mathrm{O}(n^3)$ to $\mathrm{O}(n^2)$, where $\mathrm{O}(n^2)$ represents the complexity of matrix multiplication.

Recently, a series of studies have extended the matrix sketching technique to range estimation for high-order tensor structures, such as the Tucker structure [30, 209, 154] and the tensor train structure [3, 122, 198]. These studies developed specialized structures for sketching to reduce computational complexity while maintaining high levels of numerical accuracy in handling high-order tensors. Previous work has also explored randomized sketching techniques in specific estimation problems. For instance, [148] and [181] utilized randomized sketching to solve unconstrained least squares problems. [234] , [180], [125] and [214] improved the Nyström method with randomized sketching techniques. Similarly, [4], [226], and [239] applied randomized sketching to kernel matrices in kernel ridge regression to reduce computational complexity.

Algorithms for Tensor-Train Decomposition: A variety of efficient algorithms have been developed to compute tensor-train (TT) decompositions for high-dimensional tensors. The most fundamental approach is the TT-SVD algorithm [167], which constructs the TT representation via a sequence of matricizations and truncated singular value decompositions (SVDs). This method provides a quasi-optimal low-rank approximation with controlled accuracy, but requires access to the full tensor, which may be prohibitive in large-scale settings. To alleviate this limitation, cross approximation methods such as TT-cross and adaptive cross approximation have been proposed [166, 190, 191]. These methods recover the TT representation by sampling carefully selected tensor entries, significantly reducing computational complexity while maintaining approximation quality. Closely related approaches include interpolation-based and skeleton decomposition techniques, which further improve efficiency for structured tensors.

Another important class of methods is based on iterative optimization over the manifold of fixed TT-rank tensors. Alternating least squares (ALS) and density matrix renormalization group (DMRG) algorithms [95, 192, 231] update one or two TT cores at a time while keeping

the others fixed. Variants such as alternating minimal energy methods [57] and Riemannian optimization approaches [143, 207] further improve convergence and stability, especially for solving high-dimensional linear systems and eigenvalue problems.

In addition, tensor completion and recovery methods in TT format have been studied, where the goal is to reconstruct a low-rank tensor from partial observations [74, 207]. These approaches are closely related to matrix completion and often rely on alternating optimization or Riemannian gradient methods.

Together, these methods form a comprehensive computational framework for constructing TT representations, enabling efficient approximation, optimization, and learning in high-dimensional problems.

CHAPTER 2

DENSITY ESTIMATION VIA VARIANCE REDUCED SKETCHING

2.1 Introduction

Suppose the data $\{Z_i\}_{i=1}^N \subset \mathbb{R}^d$ are independently sampled from an unknown density $p^* \in L_2(\mathbb{R}^d)$. In this work, we aim to develop a new framework specifically designed to estimate multivariate nonparametric density functions. Here we view density functions as order d tensors in the infinite-dimensional function space $L_2(\mathbb{R}^d) = L_2(\mathbb{R}) \otimes \cdots \otimes L_2(\mathbb{R})$. In the setting of estimating an α-times differentiable density function in d dimensions, our estimator achieves error rates

$$\mathbb{E}(\|p^* - \widehat{p}_{\mathrm{VRS}}\|_{L_2}) = \mathrm{O}\left(\frac{\sqrt{\prod_{j=1}^d r_j}}{N^{\alpha/(2\alpha+1)}} + \xi^*_{(r_1,\ldots,r_d)}\right). \tag{2.1}$$

Here, $\{r_j\}_{j=1}^d$ are the user-specified multilinear ranks for the estimator $\widehat{p}_{\mathrm{VRS}}$, which is an order d tensor in the function space; and $\xi^*_{(r_1,\ldots,r_d)}$ represents the best possible error one could achieve using any multilinear rank-$(r_1,\ldots,r_d)$ tensor to approximate p^* in $L_2(\mathbb{R}^d)$:

$$\xi^*_{(r_1,\ldots,r_d)} = \min_{B \text{ has ranks } (r_1,\ldots,r_d)} \{\|p^* - B\|_{L_2}\}.$$

The multilinear rank means that each mode-i flattening of B has rank r_i in function space, where this notion of rank is defined in the following Theorem 1. The definition of multilinear rank is frequently used in the literature, such as [119, 48, 221]. Moreover, the precise definition of $\xi^*_{(r_1,\ldots,r_d)}$ can be found at (2.24). In contrast, the error rate of the classical kernel

density estimator (KDE) satisfies (see e.g.[229])

$$\mathbb{E}(\|p^* - \widehat{p}_{\text{KDE}}\|_{L_2}) = \mathrm{O}\left(\frac{1}{N^{\alpha/(2\alpha+d)}}\right), \tag{2.2}$$

We refer to our approach as *Variance-Reduced Sketching* (VRS) since the theoretical result provides a reduced curse of dimensionality. In Section 2.4, we show that functions from many well-known models such as the additive model[1] and the mean-field model[2] are low-rank tensors with bounded ranks $(r_1, \ldots, r_d)$ and approximation error $\xi^*_{(r_1,\ldots,r_d)} = 0$. In such settings, the error rate in (2.1) reduces to $\mathrm{O}\left(\frac{1}{N^{2\alpha/(2\alpha+1)}}\right)$, which corresponds to the nonparametric error rate in one dimension, significantly better than (2.2).

The promising performance of Variance-Reduced Sketching (VRS) comes from its ability to reduce the problem of multivariate density estimation to the problem of estimating low-rank functional matrices/tensors using their moments. In the finite-dimensional low-rank matrix estimation setting, estimating the matrix is equivalent to estimating its range, which is determined by its singular vectors. This intuition extends to the density estimation setting. As a concrete example, suppose $p^*(x, y) : \mathbb{R}^2 \to \mathbb{R}^+$ is a two-dimensional density function. It can be viewed as an infinite-dimensional matrix in $L_2(\mathbb{R}) \otimes L_2(\mathbb{R})$. If, in addition, p^* is low-rank, then estimating the matrix p^* is equivalent to estimating its function range. Here, the function range of p^* with respect to the variable x is defined as

$$\text{Range}_x(p^*) = \left\{ \int p^*(x, y)g(y)\,\mathrm{d}y : g(y) \in L_2(\mathbb{R}) \right\}.$$

Note that the function range of $p^*(x, y)$ forms a linear subspace of functions in the variable $x \in \mathbb{R}$. The moral from linear algebra indicates that, by multiplying a low-rank matrix with some carefully chosen vectors, the range can be readily obtained. To estimate

1. $p : \mathbb{R}^d \to \mathbb{R}$ is additive if $p(z_1, \ldots, z_d) = p_1(z_1) + p_2(z_2) + \ldots + p_d(z_d)$.

2. $p : \mathbb{R}^d \to \mathbb{R}$ is mean-field if $p(z_1, \ldots, z_d) = p_1(z_1) \cdot p_2(z_2) \cdots p_d(z_d)$.

the function range, as demonstrated in Figure 2.1, in VRS we integrate p^* with carefully chosen functions, analogous to finite-dimensional matrix-vector multiplications. Integrating p^* against low-frequency functions results in moments of p^* that are highly correlated to the signals. We refer to this technique as a variance-reduced sketch for matrices in function spaces. These moments, typically of low orders, are selected to ensure that the range of the density can be accurately recovered while maintaining low variance in range estimation.

In this way, VRS transforms a multidimensional density estimation problem into the task of estimating one-dimensional moments, achieving the univariate estimation error. Additionally, this moment sketching strategy requires only a single pass over the samples and dimensions, resulting in a computational cost that scales linearly with the sample size N. The moment-based range estimation technique can be readily generalized to density estimation in arbitrary dimensions.

We stress that our VRS approach is not simply an orthonormal basis expansion followed by a standard Tucker approximation, which is the conventional pipeline for Tucker decomposition. Instead, the main idea is to estimate the column space of each mode flattening of the large tensor, without requiring full estimation of the corresponding row spaces. Because the high-dimensional tensor is implicitly defined through samples rather than stored explicitly, we can operate directly on these samples and avoid storing the full tensor. Combing this with the sketching strategy, we can tackle moderately higher-dimensional problems rather than being restricted to three-dimensional tensors, which are the typical setting for standard Tucker methods.

We emphasize that our VRS framework is fundamentally different from the randomized sketching algorithm for finite-dimensional matrices, as randomly chosen sketching does not address the curse of dimensionality in multivariable density estimation models. Additionally, the statistical guarantees for VRS are derived from a computationally tractable algorithm. In contrast, deep learning methods exhibit generalization errors that depend strongly on the

architecture of the neural network estimators. Moreover, the statistical analysis of these generalization errors is not necessarily aligned with the optimization errors achieved by computationally tractable optimizers.

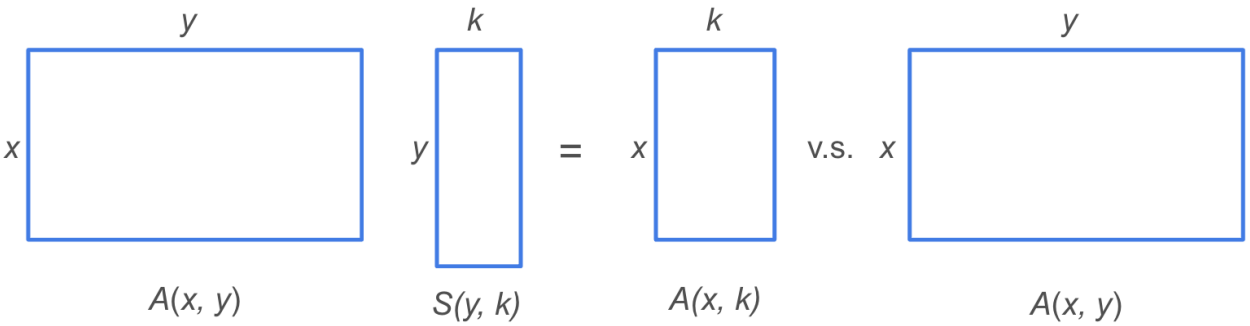


Figure 2.1: The sketched function AS by VRS retains the range in the variable x of $A(x, y)$. The complexity of estimating the range of A using AS is much lower than the complexity of directly estimating A.

The rest of work is organized as follows. In Section 2.2, we detail the procedure for implementing the range estimation through sketching and study the corresponding error analysis. In Section 2.3, we extend our study to density function estimation based on the range estimators developed in Section 2.2, and provide the corresponding statistical error analysis with a reduced curse of dimensionality. In Section 2.5, we present comprehensive numerical results to demonstrate the superior numerical performance of our method. Finally, Section 2.6 summarizes our conclusions.

Related Literature

Matrix approximation algorithms, such as singular value decomposition and QR decomposition, play a crucial role in computational mathematics and statistics. A notable advancement in this area is the emergence of randomized low-rank approximation algorithms. These algorithms excel in reducing time and space complexity substantially without sacrificing too much numerical accuracy. Seminal contributions to this area are outlined in

works such as [133] and [82]. Additionally, review papers like [235], [59], [149], and [150] have provided comprehensive summaries of these randomized approaches, along with their theoretical stability guarantees. Randomized low-rank approximation algorithms typically start by estimating the range of a large low-rank matrix $p \in \mathbb{R}^{n \times n}$ by forming a reduced-size sketch. This is achieved by right multiplying p with a random matrix $S \in \mathbb{R}^{n \times k}$, where $k \ll n$. The random matrix S is selected to ensure that the range of pS remains a close approximation of the range of p, even when the column size of pS is significantly reduced from p. As such, the random matrix S is referred to as randomized linear embedding or sketching matrix by [215] and [156]. The sketching approach reduces the cost in singular value decomposition from $\mathrm{O}(n^3)$ to $\mathrm{O}(n^2)$, where $\mathrm{O}(n^2)$ represents the complexity of matrix multiplication.

Recently, a series of studies have extended the matrix sketching technique to range estimation for high-order tensor structures, such as the Tucker structure [30, 209, 154] and the tensor train structure [3, 122, 198]. These studies developed specialized structures for sketching to reduce computational complexity while maintaining high levels of numerical accuracy in handling high-order tensors.

Previous work has also explored randomized sketching techniques in specific estimation problems. For instance, [148] and [181] utilized randomized sketching to solve unconstrained least squares problems. [234] , [180], [125] and [214] improved the Nyström method with randomized sketching techniques. Similarly, [4], [226], and [239] applied randomized sketching to kernel matrices in kernel ridge regression to reduce computational complexity. While these studies mark significant progress in the literature, they usually require extensive sampling or evaluation of the estimated function to employ the randomized sketching technique, in order to maintain acceptable accuracy. This step would be significantly expensive for the higher-dimensional setting. Notably, [103] and subsequent studies [210, 183, 175, 37, 211] addressed these issues by incorporating the variance of the data generation process into the design of

sketching techniques for high-dimensional tensor estimation. This sketching technique allows for the direct estimation of the range of a tensor with reduced sample complexity, rather than directly estimating the full tensor.

In related work on Tucker decomposition, higher-order singular value decomposition (HOSVD), also known as Tucker decomposition [189, 120, 13, 30, 154, 1], has achieved notable success across a wide range of applications. However, most existing approaches are most effective in three-dimensional tensor settings; extending them to moderately higher-order tensors can be computationally prohibitive without additional techniques to mitigate the resulting complexity. In contrast, our work aims to recast a multi-dimensional density estimation problem as the task of estimating a collection of one-dimensional moments, which enables us to handle general problems in high dimensions.

Notations

Let $\mathbb{N} = \{1, 2, 3, \ldots\}$ and $\mathbb{R}$ denote the set of all the real numbers. We denote $X_n = \mathrm{O}_{\mathbb{P}}(a_n)$ if for any given $\epsilon > 0$, there exists a $K_\epsilon > 0$ such that $\limsup_{n\to\infty} \mathbb{P}(|X_n/a_n| \geq K_\epsilon) < \epsilon$. For real numbers $\{a_n\}_{n=1}^{\infty}$, $\{b_n\}_{n=1}^{\infty}$ and $\{c_n\}_{n=1}^{\infty}$, we denote $a_n = \mathrm{O}(b_n)$ if $\lim_{n\to\infty} a_n/b_n < \infty$ and $a_n = \mathrm{o}(c_n)$ if $\lim_{n\to\infty} a_n/c_n = 0$. Let $[0, 1]$ denote the unit interval in $\mathbb{R}$. For $d \in \mathbb{N}$, denote the d-dimensional unit cube as $[0, 1]^d$.

Let $\{f_i\}_{i=1}^n$ be a collection of elements in the Hilbert space $\mathcal{H}$. Then

$$\mathrm{Span}\{f_i\}_{i=1}^n = \{b_1 f_1 + \cdots + b_n f_n : \{b_i\}_{i=1}^n \subset \mathbb{R}\}. \tag{2.3}$$

Note that $\mathrm{Span}\{f_i\}_{i=1}^n$ is a linear subspace of $\mathcal{H}$.

For a generic measurable set $\Omega \subset \mathbb{R}^d$, denote $L_2(\Omega) = \big\{f : \Omega \to \mathbb{R} : \|f\|^2_{L_2(\Omega)} = \int_\Omega f^2(z)\mathrm{d}z < \infty\big\}$. For any $f, g \in L_2(\Omega)$, let the inner product between f and g be $\langle f, g\rangle = \int_\Omega f(z)g(z)\mathrm{d}z$. We say that $\{\phi_k\}_{k=1}^{\infty}$ is a collection of orthonormal functions in $L_2(\Omega)$ if

$\langle \phi_k, \phi_l \rangle = \delta_{kl}$. We say that $\{\phi_k\}_{k=1}^{\infty}$ is a complete basis of $L_2(\Omega)$ if $\{\phi_k\}_{k=1}^{\infty}$ is orthonormal and $\text{Span}\{\phi_k\}_{k=1}^{\infty} = L_2(\Omega)$. Let $Z \in \Omega$ and δ_Z be the indicator function at the point Z. Define $\langle \delta_Z, f \rangle = f(Z)$. In what follows, we briefly introduce the notation for Sobolev spaces. Let $\Omega \subset \mathbb{R}^d$ be any measurable set. For multi-index $\beta = (\beta_1, \ldots, \beta_d) \in \mathbb{N}^d$, and $f(z_1, \ldots, z_d) : \Omega \to \mathbb{R}$, define the β-derivative of f as $D^\beta f = \partial_1^{\beta_1} \cdots \partial_d^{\beta_d} f$. Then

$$W_2^\alpha(\Omega) := \{f \in L_2(\Omega) : \ D^\beta f \in L_2(\Omega) \text{ for all } |\beta| \leq \alpha\}, \tag{2.4}$$

where $|\beta| = \beta_1 + \cdots + \beta_d$ and $\alpha \in \mathbb{N}$ represents the total order of derivatives. The Sobolev norm of $f \in W_2^\alpha$ is $\|f\|^2_{W_2^\alpha(\Omega)} = \sum_{0 \leq |\beta| \leq \alpha} \|D^\beta f\|^2_{L_2(\Omega)}$.

Linear algebra in tensorized function spaces is the key component to develop our Variance-Reduced Sketching (VRS) framework for nonparametric density estimation. We start with a convenient notation for partial integration of two functions. For $j = 1, 2, 3$ let $\Omega_j \subset \mathbb{R}^{d_j}$ be any regular domain. Let $p(x, y) \in L_2(\Omega_1 \times \Omega_2)$ and $Q(y, z) \in L_2(\Omega_2 \times \Omega_3)$ be two functions. Define

$$(p \times_y Q)(x, z) = \int_{\Omega_2} p(x, y) Q(y, z) \mathrm{d}y. \tag{2.5}$$

For $j \in \{1, \ldots, d\}$, let $\Omega_j \subset \mathbb{R}^{d_j}$ be a measurable subset. Let $\mathcal{S}_j$ be arbitrary linear subspaces of $L_2(\Omega_j)$ where functions in $\mathcal{S}_j$ correspond to the variable z_j. Define

$$\mathcal{S}_1 \otimes \cdots \otimes \mathcal{S}_d = \text{Span}\{g_1(z_1) \cdots g_d(z_d) : g_1(z_1) \in \mathcal{S}_1, \ldots, g_d(z_d) \in \mathcal{S}_d\}.$$

So $\mathcal{S}_1 \otimes \cdots \otimes \mathcal{S}_d \subset L_2(\Omega_1) \otimes \cdots \otimes L_2(\Omega_d) = L_2(\Omega_1 \times \cdots \times \Omega_d)$. Let $\{\phi_l^{(j)}(z_j)\}_{l=1}^{\infty}$ be a complete basis in $L_2(\Omega_j)$, and define $\mathcal{M}_j = \text{Span}\{\phi_l^{(j)}(z_j)\}_{l=1}^{m}$. So $\mathcal{M}_j \subset L_2(\Omega_j)$ and

$\mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d \subset L_2(\Omega_1) \otimes \cdots \otimes L_2(\Omega_d)$. Denote

$$\mathcal{P}_{\mathcal{M}_j}(z_j, z_j') = \sum_{l=1}^{m} \phi_l^{(j)}(z_j)\phi_l^{(j)}(z_j').$$

For any function $p \in L_2(\Omega_1 \times \cdots \times \Omega_d)$, we define $p \times_j \mathcal{P}_{\mathcal{M}_j}(z_1, \ldots, z_d)$ using (2.5) as

$$p \times_1 \mathcal{P}_{\mathcal{M}_1}(z_1, \ldots, z_d) = \int_{\mathcal{O}} p(z_1, \ldots, z_{j-1}, z_j', z_{j+1}, \ldots, z_d)\mathcal{P}_{\mathcal{M}_j}(z_j', z_j)\mathrm{d}z_j'.$$

Therefore $p \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}$ can be defined inductively. A direct calculation shows that

$$p \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}(z_1, \ldots, z_d) = \sum_{l_1=1}^{m} \cdots \sum_{l_d=1}^{m} \langle p, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle \phi_{l_1}(z_1) \cdots \phi_{l_d}(z_d).$$

Here

$$\langle p, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle = \int \cdots \int p(z_1, \ldots, z_d)\phi_{l_1}(z_1) \cdots \phi_{l_d}(z_d)\mathrm{d}z_1 \cdots \mathrm{d}z_d.$$

Therefore $p \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}$ is a function in the space $\mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d$. Given data $\{Z_i\}_{i=1}^N \subset \Omega_1 \times \cdots \times \Omega_d$, denote the corresponding empirical measure as $\widehat{p} = \frac{1}{N}\sum_{i=1}^N \delta_{Z_i}$, where δ_{Z_i} is the indicator function at the point Z_i. We can also project the empirical measure $\widehat{p}$ onto $\mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d$ as

$$\begin{aligned}
\widehat{p} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}(z_1, \ldots, z_d) &= \sum_{l_1=1}^{m} \cdots \sum_{l_d=1}^{m} \langle \widehat{p}, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle \phi_{l_1}(z_1) \cdots \phi_{l_d}(z_d) \\
&= \sum_{l_1=1}^{m} \cdots \sum_{l_d=1}^{m} \left\{ \frac{1}{N} \sum_{i=1}^{N} \phi_{l_1} \otimes \cdots \otimes \phi_{l_d}(Z_i) \right\} \phi_{l_1}(z_1) \cdots \phi_{l_d}(z_d).
\end{aligned} \tag{2.6}$$

2.2 Density Range Estimation by Sketching

In this section, we introduce a new technique called sketching, which allows us to estimate the range of a density function based on discrete samples with a reduced curse of dimensionality. The definition of function range can be found in (2.7). As demonstrated in Section 2.3, range estimation plays a crucial role in the VRS density estimation framework.

Throughout this section, we set d_1, d_2 to be two arbitrary positive integers. Let $\Omega_1 \subset \mathbb{R}^{d_1}$ and $\Omega_2 \subset \mathbb{R}^{d_2}$ be two measurable sets. Let $p^*(x, y) : \Omega_1 \times \Omega_2 \to \mathbb{R}$ be the unknown density function.

Let $B(x, y) \in L_2(\Omega_1 \times \Omega_2)$. We define the range space of $B(x, y)$ in the variable x as

$$\text{Range}_x(B) = \left\{ f(x) : f(x) = \int_{\Omega_2} B(x, y) g(y) \mathrm{d}y \text{ for any } g(y) \in L_2(\Omega_2) \right\}. \tag{2.7}$$

In VRS, we first estimate the range of the unknown density p^* with respect to each variable, and then incorporate the range estimators of all variables to estimate the density function p^*.

We define the dimensionality of B with respect to variables $(x, y) \in \Omega_1 \times \Omega_2$ by using the singular value decomposition of B in the function space $L_2(\Omega_1 \times \Omega_2)$.

Theorem 1. *[Singular value decomposition in function space] Let $B(x, y) : \Omega_1 \times \Omega_2 \to \mathbb{R}$ be any function such that $\|B\|_{L_2(\Omega_1 \times \Omega_2)} < \infty$. There exists a collection of strictly positive singular values $\{\sigma_\rho(B)\}_{\rho=1}^r \subset \mathbb{R}^+$ in decreasing order, and two collections of orthonormal basis functions $\{\Phi_\rho(x)\}_{\rho=1}^r \subset L_2(\Omega_1)$ and $\{\Psi_\rho(y)\}_{\rho=1}^r \subset L_2(\Omega_2)$, where $r \in \mathbb{N} \cup \{+\infty\}$, such that*

$$B(x, y) = \sum_{\rho=1}^{r} \sigma_\rho(B)\, \Phi_\rho(x)\, \Psi_\rho(y). \tag{2.8}$$

In this case, we say that the rank of $B(x, y)$ is r.

Suppose (2.8) holds. An equivalent definition of $\text{Range}_x(B)$ is

$$\text{Range}_x(B) = \text{Span}\{\Phi_\rho(x)\}_{\rho=1}^r.$$

Therefore, $\dim(\text{Range}_x(B))$ equals the rank of B. This illustrates the well-known fact that the dimensionality of the column space of a matrix agrees with its rank.

Let $\{Z_i\}_{i=1}^N \subset \Omega_1 \times \Omega_2$ be the observed data sampled from $p^*(x,y)$. Based on the observed data, our goal in this section is to estimate $\text{Range}_x(p^*)$ defined in (2.7). We denote by $\widehat{p}$ the empirical measure formed by $\{Z_i\}_{i=1}^N \subset \mathbb{R}^d$ as

$$\widehat{p}(x,y) = \frac{1}{N}\sum_{i=1}^N \delta_{Z_i}(x,y). \tag{2.9}$$

We will work with two user-specified linear subspaces $\mathcal{M}$ and $\mathcal{S}$, where $\mathcal{M} \subset L_2(\Omega_1)$ serves as an estimation subspace, and $\mathcal{S} \subset L_2(\Omega_2)$ serves as a sketching subspace. Suppose that

$$\mathcal{M} = \text{Span}\{v_l(x)\}_{l=1}^{\dim(\mathcal{M})} \quad \text{and} \quad \mathcal{S} = \text{Span}\{w_\eta(y)\}_{\eta=1}^{\dim(\mathcal{S})}, \tag{2.10}$$

where $\{v_l(x)\}_{l=1}^{\dim(\mathcal{M})}$ are orthogonal functions in $L_2(\Omega_1)$ and $\{w_\eta(y)\}_{\eta=1}^{\dim(\mathcal{S})}$ are orthogonal functions in $L_2(\Omega_2)$. Specific choices of these subspaces can be found at (2.16) and (2.17). Our procedure is composed of the following three stages.

- Sketching stage. We apply the projection operator $\mathcal{P}_\mathcal{S}$ to $\widehat{p}$ by computing

$$\left\{\int_{\Omega_2} \widehat{p}(x,y) w_\eta(y) \mathrm{d}y\right\}_{\eta=1}^{\dim(\mathcal{S})}. \tag{2.11}$$

  Note that for each $\eta = 1, \ldots, \dim(\mathcal{S})$, $\int_{\Omega_2} \widehat{p}(x,y) w_\eta(y) \mathrm{d}y$ is a function solely that depends on x. This stage aims at reducing the curse of dimensionality associated to variable y.

- Estimation stage. We estimate the functions $\left\{ \int_{\Omega_2} \widehat{p}(x,y) w_\eta(y) \mathrm{d}y \right\}_{\eta=1}^{\dim(\mathcal{S})}$ by utilizing the estimation space $\mathcal{M}$. Specifically, for each $\eta = 1, \ldots, \dim(\mathcal{S})$, we approximate $\int_{\Omega_2} \widehat{p}(x,y) w_\eta(y) \mathrm{d}y$ by

$$\widetilde{f}_\eta(x) = \arg\min_{f \in \mathcal{M}} \left\| \int_{\Omega_2} \widehat{p}(x,y) w_\eta(y) \mathrm{d}y - f(x) \right\|_{L_2(\Omega_1)}^2. \tag{2.12}$$

- Orthogonalization stage. Let

$$\widetilde{p}(x,y) = \sum_{\eta=1}^{\dim(\mathcal{S})} \widetilde{f}_\eta(x) w_\eta(y). \tag{2.13}$$

Compute the leading singular functions in the variable x of $\widetilde{p}(x,y)$ to estimate the $\mathrm{Range}_x(p^*)$.

We formally summarize our procedure in Algorithm 1.

Algorithm 1 Density range Estimation via Variance-Reduced Sketching

INPUT: Empirical measure $\widehat{p} = \frac{1}{N} \sum_{i=1}^N \delta_{Z_i}$, parameter $r \in \mathbb{Z}^+$, linear subspaces $\mathcal{M} = \mathrm{Span}\{v_l(x)\}_{l=1}^{\dim(\mathcal{M})}$ and $\mathcal{S} = \mathrm{Span}\{w_\eta(y)\}_{\eta=1}^{\dim(\mathcal{S})}$.

1: Compute $\left\{ \int_{\Omega_2} \widehat{p}(x,y) w_\eta(y) \mathrm{d}y \right\}_{\eta=1}^{\dim(\mathcal{S})}$, the projection of $\widehat{p}$ onto $\{w_\eta(y)\}_{\eta=1}^{\dim(\mathcal{S})}$.

2: Compute the estimated functions $\{\widetilde{f}_\eta(x)\}_{\eta=1}^{\dim(\mathcal{S})}$ in $\mathcal{M}$ by (2.12).

3: Compute the leading r singular functions in the variable x of $\widetilde{p}(x,y) = \sum_{\eta=1}^{\dim(\mathcal{S})} \widetilde{f}_\eta(x) w_\eta(y)$ and denote them as $\{\widehat{\Phi}_\rho(x)\}_{\rho=1}^r$.

OUTPUT: $\widehat{\mathcal{P}}_x(x,x') = \sum_{\rho=1}^r \widehat{\Phi}_\rho(x) \widehat{\Phi}_\rho(x')$.

An explicit expression of $\widetilde{p}(x,y)$ in (2.13) can be obtained from $\widehat{p}(x,y)$. More precisely, in the sketching stage, (2.11) is equivalent to computing $\widehat{p} \times_y \mathcal{P}_{\mathcal{S}}$. Indeed, by (2.6),

$$\widehat{p} \times_y \mathcal{P}_{\mathcal{S}} = \sum_{\eta=1}^{\dim(\mathcal{S})} \left(\int_{\Omega_2} \widehat{p}(x,y) w_\eta(y) \mathrm{d}y \right) w_\eta(y).$$

In the estimation stage, using orthogonality of $\{v_l(x)\}_{l=1}^{\dim(\mathcal{M})}$, (2.12) can be explicitly ex-

pressed as

$$\widehat{f}_\eta(x) = \sum_{l=1}^{\dim(\mathcal{M})} \langle \widehat{p}, v_l \otimes w_\eta \rangle v_l(x),$$

where by definition, $\langle \widehat{p}, v_l \otimes w_\eta \rangle = \int_{\Omega_2} \int_{\Omega_1} \widehat{p}(x,y) v_l(x) w_\eta(y) \mathrm{d}x \mathrm{d}y$. Therefore, $\widetilde{p}(x,y)$ in (2.13) can be rewritten as

$$\widetilde{p}(x,y) = \sum_{l=1}^{\dim(\mathcal{M})} \sum_{\eta=1}^{\dim(\mathcal{S})} \langle \widehat{p}, v_l \otimes w_\eta \rangle v_l(x) w_\eta(y). \tag{2.14}$$

By (2.6), $\widehat{p} \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{L}$ has the exact same expression as $\widetilde{p}$. Therefore, we establish the identification

$$\widetilde{p} = \widehat{p} \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{L}. \tag{2.15}$$

We discuss the specific choices of $\mathcal{M}$ and $\mathcal{S}$. Let $\mathcal{O}$ be a measurable subset of $\mathbb{R}$. Suppose that $\Omega_1 \subset \mathcal{O}^{d_1}$ and $\Omega_2 \subset \mathcal{O}^{d_2}$. In density estimation, it is sufficient to assume $\mathcal{O} = [0,1]$. Indeed, if the density function p^* has compact support, through necessary scaling, we can assume the support $\Omega_1 \times \Omega_2$ is a subset of $\mathcal{O}^{d_1} \times \mathcal{O}^{d_2} = [0,1]^{d_1+d_2}$.

Let $\{\phi_k\}_{k=1}^\infty \subset L_2(\mathcal{O})$ be a collection of orthonormal one-dimensional basis functions generated from either the reproducing kernel Hilbert spaces or the Legendre polynomial system. A detailed discussion of about the reproducing kernel Hilbert spaces or the Legendre polynomial system can be found in Section A.5. Let $x_j \in \mathbb{R}$ for $j \in \{1, \ldots, d_1\}$ and let $y_k \in \mathbb{R}$ for $k \in \{1, \ldots, d_2\}$. For $m, \ell \in \mathbb{N}$, let

$$\mathcal{M} = \mathrm{span}\left\{ \phi_{l_1}(x_1)\phi_{l_2}(x_2)\cdots\phi_{l_{d_1}}(x_{d_1}) \right\}_{l_1,\ldots,l_{d_1}=1}^{m} \quad \text{and} \tag{2.16}$$

$$\mathcal{S} = \mathrm{span}\left\{ \phi_{\eta_1}(y_1)\phi_{\eta_2}(y_2)\cdots\phi_{\eta_{d_2}}(y_{d_2}) \right\}_{\eta_1,\ldots,\eta_{d_2}=1}^{\ell}. \tag{2.17}$$

Therefore $\dim(\mathcal{M}) = m^{d_1}$ and $\dim(\mathcal{S}) = \ell^{d_2}$. In the following result, we show that the range estimator in Algorithm 1 is consistent.

Theorem 2. *Suppose that the rank of p^* is $r < \infty$, $p^* \in W_2^\alpha(\mathcal{O}^d)$, and let $\sigma_r^* = \sigma_r(p^*(x,y))$ denote the minimal nonzero singular value*[3] *of $p^*(x,y)$. In Algorithm 1, let $\mathcal{M}$ and $\mathcal{S}$ be defined as in* (2.16) *and* (2.17)*. For a sufficiently large constant C, suppose that*

$$\sigma_r^* > C \max\left\{\ell^{-\alpha}, m^{-\alpha}, \sqrt{\frac{(m^{d_1} + \ell^{d_2})\log(N)}{N}}\right\}. \tag{2.18}$$

Let $\mathcal{P}_x^*$ be the projection function onto $\mathrm{Range}_x(p^*)$*, and let $\widehat{\mathcal{P}}_x$ be the output of Algorithm 1. Then*

$$\|\widehat{\mathcal{P}}_x - \mathcal{P}_x^*\|_{L_2(\Omega_1\times\Omega_1)}^2 = \mathrm{O}_{\mathbb{P}}\left\{\frac{r}{(\sigma_r^*)^2}\left(m^{-2\alpha} + \frac{(m^{d_1} + \ell^{d_2})\log(N)}{N}\right)\right\}. \tag{2.19}$$

The proof of Theorem 2 can be found in Section A.2. It follows that if $m \asymp N^{1/(2\alpha+d_1)}$, $\ell = C_L(\sigma_r^*)^{-1/\alpha}$ for a sufficiently large constant C_L, and sample size follows the inequality $N \geq C_\sigma \max\{(\sigma_r^*)^{-2-d_1/\alpha}, (\sigma_r^*)^{-2-d_2/\alpha}\}$ for a sufficiently large constant C_σ, then Theorem 2 implies that up to a logarithm factor,

$$\|\widehat{\mathcal{P}}_x - \mathcal{P}_x^*\|_{L_2(\Omega_1\times\Omega_1)}^2 = \mathrm{O}_{\mathbb{P}}\left\{\frac{r}{N^{2\alpha/(2\alpha+d_1)}(\sigma_r^*)^2} + \frac{r}{N(\sigma_r^*)^{2+d_2/\alpha}}\right\}. \tag{2.20}$$

In Section 2.4, we demonstrate that both additive functions and the mean-field functions are finite-rank functions with minimal singular values being strictly positive. In these settings, σ_r^* and r are both absolute constants, and up to a logarithm factor, (2.20) further reduces to,

$$\|\widehat{\mathcal{P}}_x - \mathcal{P}_x^*\|_{L_2(\Omega_1\times\Omega_1)}^2 = \mathrm{O}_{\mathbb{P}}\left\{\frac{1}{N^{2\alpha/(2\alpha+d_1)}}\right\},$$

3. The definition of the singular values of a two-variable function can be found in (2.8).

which matches the optimal nonparametric density estimation rate in d_1 dimensions. This indicates that our sketching method is able to estimate $\text{Range}_x(p^*)$ without the curse of dimensionality introduced by the variable $y \in \mathbb{R}^{d_2}$. Utilizing the sketching estimators, we can further estimate the unknown function p^* with a reduced curse of dimensionality as detailed in Section 2.3.

2.3 Density Estimation by Sketching

In this section, we propose a new multi-dimensional density estimator by utilizing the range estimator outlined in Algorithm 1. Let $\mathcal{O} \subset \mathbb{R}$ be a measurable set and $p^*(z_1, \ldots, z_d) : \mathcal{O}^d \to \mathbb{R}$ be the unknown population function. In density estimation, it is sufficient to assume $\mathcal{O} = [0, 1]$. Indeed, if p^* has compact support, through necessary scaling, we can assume the support is a subset of $\mathcal{O}^d = [0, 1]^d$.

In Algorithm 2, we formally summarize our tensor-based estimator of p^*. The selection of tuning parameters in Algorithm 2 will be discussed in the numerical section.

We formally introduce the estimation and the sketching spaces for VRS. Let $\{\phi_k\}_{k=1}^{\infty} \subset L_2(\mathcal{O})$ be a collection of one-dimensional orthonormal basis functions. For $j \in \{1, \ldots, d\}$, let $m \in \mathbb{N}, \ell_j \in \mathbb{N}$ and denote

$$\mathcal{M}_j = \text{span}\left\{\phi_l(z_j)\right\}_{l=1}^{m} \quad \text{and} \tag{2.21}$$

$$\mathcal{S}_j = \text{span}\left\{\phi_{\eta_1}(z_1) \cdots \phi_{\eta_{j-1}}(z_{j-1})\phi_{\eta_{j+1}}(z_{j+1}) \cdots \phi_{\eta_d}(z_d)\right\}_{\eta_1,\ldots,\eta_{j-1},\eta_{j+1},\ldots,\eta_d=1}^{\ell_j}. \tag{2.22}$$

Assumption 1. *Let $\{\phi_l\}_{l=1}^{\infty}$ be a collection of one-dimensional basis functions in $L_2(\mathcal{O})$ generated from either the reproducing kernel Hilbert spaces or the Legendre polynomials system.*

Reproducing kernel Hilbert spaces and Legendre polynomials are widely used numerical techniques in the machine learning literature. Next, we summarize the regularity condition

Algorithm 2 Multivariable Density Estimation via Variance-Reduced Sketching

INPUT: Data $\{Z_i\}_{i=1}^N$, parameters $\{r_j\}_{j=1}^d \subset \mathbb{Z}^+$, estimation subspaces $\{\mathcal{M}_j\}_{j=1}^d$ as in (2.21) and sketching subspaces $\{\mathcal{S}_j\}_{j=1}^d$ as in (2.22).

Set empirical measures $\widehat{p} = \frac{1}{N/2}\sum_{i=1}^{N/2} \delta_{Z_i}$, and $\widehat{p}' = \frac{1}{N/2}\sum_{i=N/2+1}^{N} \delta_{Z_i}$

[**Subspaces estimation via Sketching**]

for $j \in \{1, \ldots, d\}$ **do**

Set $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$.

Compute the leading r_j singular functions in the variable z_j of the function

$$\left(\widehat{p} \times_j \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j}\right)(z_j, y),$$

and denote them as $\{\widehat{\Phi}_\rho(z_j)\}_{\rho=1}^{r_j}$.

Set the projection function $\mathcal{P}_j^{\Phi}(z_j, z_j') = \sum_{\rho=1}^{r_j} \widehat{\Phi}_\rho(z_j)\widehat{\Phi}_\rho(z_j')$.

end for

[**Sketching using estimated subspaces**]

for $j \in \{1, \ldots, d\}$ **do**

Set

$$\widehat{B}_j = \widehat{p} \times_j \mathcal{P}_{\mathcal{M}_j} \times_1 \mathcal{P}_1^{\Phi} \times \cdots \times_{j-1} \mathcal{P}_{j-1}^{\Phi} \times_{j+1} \mathcal{P}_{j+1}^{\Phi} \cdots \times_d \mathcal{P}_d^{\Phi}.$$

Set $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$.

Compute the leading r_j singular functions in the variable z_j of the function $\widehat{B}_j(z_j, y)$, and

denote them as $\{\widehat{\Psi}_\rho(z_j)\}_{\rho=1}^{r_j}$.

Set the projection function $\mathcal{P}_j^{\Psi}(z_j, z_j') = \sum_{\rho=1}^{r_j} \widehat{\Psi}_\rho(z_j)\widehat{\Psi}_\rho(z_j')$.

end for

Compute the estimated density function

$$\widetilde{p}(z_1, \ldots, z_d) = \left(\widehat{p}' \times_1 \mathcal{P}_1^{\Psi} \times \cdots \times_d \mathcal{P}_d^{\Psi}\right)(z_1, \ldots, z_d).$$

OUTPUT: The estimated density function $\widetilde{p}(z_1, \ldots, z_d)$.

used to establish the consistency of VRS.

Assumption 2. *Suppose that the data* $\{Z_i\}_{i=1}^N \subset \mathcal{O}^d$ *are independently sampled from the density* $p^* : \mathcal{O}^d \to \mathbb{R}^+$. *Suppose in addition that* $\|p^*\|_{W_2^\alpha(\mathcal{O}^d)} < \infty$ *with* $\alpha \geq 1$ *and that* $\|p^*\|_\infty < \infty$.

Assumption 2 is widely used in the density estimation literature. We aim at establishing the consistency of VRS even when the underlying density function is infinite-rank. This leads us to assume the following singular-gap condition for p^*.

Assumption 3. *For $j = 1, \dots, d$, denote $y_j = (z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_d)$, and $\sigma^*_{j,\rho} = \sigma_\rho(p^*(z_j, y_j))$ for any $\rho \in \mathbb{N}$. Here $\sigma_\rho(p^*(z_j, y_j))$ is the ρ-th singular value of the two-variable function $p^*(z_j, y)$. Suppose that*

$$N/\log(N) > C_{snr}\big(\sigma^*_{j,r_j}\big)^{\frac{-d+1}{\alpha}-2} \quad \textit{and} \quad \sigma^*_{j,r_j+1} \le \sigma^*_{j,r_j}/N^{\alpha/(2\alpha+1)}, \tag{2.23}$$

where C_{snr} is a sufficiently large constant independent of N.

Assumption 3 postulates that when we reshape p^* into the two-variable function $p^*(z_j, y_j)$, the singular gap between its r_j-th and (r_j+1)-th singular values is sufficiently large. Note that Assumption 3 trivially holds for additive functions and mean-field functions.

Finally, we quantify the oracle approximation error when finite-rank tensors are used to approximate densities that may be infinite-rank. Denote

$$\xi^*_{(r_1,\dots,r_d)} = \inf_{\substack{\dim(\mathrm{Range}_{z_j}(B)) \le r_j \\ \text{for all } j \in \{1,\dots,d\}}} \|p^* - B\|_{L_2(\mathcal{O}^d)}. \tag{2.24}$$

In other words, $\xi^*_{(r_1,\dots,r_d)}$ represents the best possible error one could achieve using any rank-$(r_1, \dots, r_d)$ tensor approximation of p^* in $L_2(\mathcal{O}^d)$. In the following result, we show that the density estimator in Algorithm 2 is consistent.

Theorem 3. *Suppose Assumption 1, Assumption 2 and Assumption 3 hold. In Algorithm 2, suppose that the inputs satisfy $\ell_j = C_L(\sigma^*_{j,r_j})^{-1/\alpha}$ for some sufficiently large constant C_L*

and $m \asymp N^{1/(2\alpha+1)}$*, and the output is* $\widetilde{p}$*. Then it holds that*

$$\|\widetilde{p} - p^*\|_{L_2(\mathcal{O}^d)} = \mathrm{O}_{\mathbb{P}}\left(\frac{\sqrt{\prod_{j=1}^d r_j}}{N^{\alpha/(2\alpha+1)}} + \xi^*_{(r_1,\ldots,r_d)}\right). \tag{2.25}$$

The proof of Theorem 3 can be found in Section A.3. In Section 2.4, we show that functions from many well-known models such as the additive model and the mean-field model are low-rank tensors with bounded ranks $(r_1, \ldots, r_d)$ and approximation error $\xi^*_{(r_1,\ldots,r_d)} = 0$. In these settings, (2.25) reduces to

$$\|\widetilde{p} - p^*\|^2_{L_2(\mathcal{O}^d)} = \mathrm{O}_{\mathbb{P}}\left(\frac{1}{N^{2\alpha/(2\alpha+1)}}\right),$$

which matches the minimax optimal rate of estimating density functions in one dimension.

2.4 Examples of Low-rank Functions

In this section, we present three examples commonly encountered in nonparametric statistical literature that are approximately low rank.

Example 1 (Additive models in regression)**.** *In multivariate nonparametric statistical literature, it is commonly assumed that the underlying unknown nonparametric function* $f^* : [0,1]^d \to \mathbb{R}$ *process an additive structure, meaning that there exists a collection of univariate functions* $\{f_j^*(z_j) : [0,1] \to \mathbb{R}\}_{j=1}^d$ *such that*

$$f^*(z_1, \ldots z_d) = f_1^*(z_1) + \cdots + f_d^*(z_d) \text{ for all } (z_1, \ldots, z_d) \in [0,1]^d$$

Let $y = (z_2, \ldots, z_d)$*, we show that the rank of* $f^*(z_1, y)$ *is at most 2. We know* $\mathrm{Range}_1(f^*) = \mathrm{Span}\{1, f_1^*(z_1)\}$*. The dimensionality of* $\mathrm{Range}_1(f^*)$ *is at most* 2*, and consequently the rank of* $f^*(z_1, y) \in L_2([0,1]) \otimes L_2([0,1]^{d-1})$ *is at most* 2*.*

Example 2 (Mean-field models in density estimation)**.** *Mean-field theory is a popular framework in computational physics and Bayesian probability as it studies the behavior of high-dimensional stochastic models. The main idea of the mean-field theory is to replace all interactions to any one body with an effective interaction in a physical system. Specifically, the mean-field model assumes that the density function* $p^*(z_1, \ldots, z_d) : [0,1]^d \to \mathbb{R}^+$ *can be well-approximated by* $p_1^*(z_1) \cdots p_d^*(z_d)$*, where for* $j = 1, \ldots, d$*,* $p_j^*(z_j) : [0,1] \to \mathbb{R}^+$ *are univariate marginal density functions. The readers are referred to [17] for further discussion. For simplicity, suppose*

$$p^*(z_1, \ldots, z_d) = p_1^*(z_1) \cdots p_d^*(z_d).$$

Then according to (2.7)*,* $\text{Range}_1(p^*) = \text{Span}\{p_1^*(z_1)\}$*. The dimensionality of* $\text{Range}_1(p^*)$ *is at most* α*, and therefore the rank of* p^* *in the variable* z_1 *is 1.*

Example 3 (Multivariate Taylor expansion)**.** *Suppose* $G : [0,1]^d \to \mathbb{R}$ *is an* α*-times continuously differentiable function. Then Taylor's theorem in the multivariate setting states that for* $z = (z_1, \ldots, z_d) \in \mathbb{R}^d$ *and* $t = (t_1, \ldots, t_d) \in \mathbb{R}^d$*,* $G(z) \approx T_t(z)$*, where*

$$T_t(z) = G(t) + \sum_{k=1}^{\alpha} \frac{1}{k!} \mathcal{D}^k G(t, z - t), \tag{2.26}$$

and $\mathcal{D}^k G(x,h) = \sum_{i_1,\cdots,i_k=1}^d \partial_{i_1} \cdots \partial_{i_k} G(x) h_{i_1} \cdots h_{i_k}$*. For example,* $\mathcal{D}G(x,h) = \sum_{i=1}^d \partial_i G(x) h_i$*,* $\mathcal{D}^2 G(x,h) = \sum_{i=1}^d \sum_{j=1}^d \partial_i \partial_j G(x) h_i h_j$*, and so on. To simplify our discussion, let* $t = 0 \in \mathbb{R}^d$*. Then* (2.26) *becomes*

$$T_0(z) = G(0) + \sum_{i=1}^d \partial_i G(0) z_i + \frac{1}{2!} \sum_{i=1}^d \sum_{j=1}^d \partial_i \partial_j G(0) z_i z_j + \ldots + \frac{1}{\alpha!} \sum_{i_1,\ldots,i_\alpha=1}^d \partial_{i_1} \cdots \partial_{i_\alpha} G(0) z_{i_1} \cdots z_{i_\alpha}.$$

Let $y = (z_2, \ldots, z_d)$*. Then by* (2.7)*,* $\text{Range}_1(T_0) = \text{Span}\{1, z_1, z_1^2, \ldots, z_1^\alpha\}$*. The dimensionality of* $\text{Range}_1(T_0)$ *is at most* $\alpha + 1$*, and therefore* G *can be well-approximated by finite rank functions.*

2.5 Numerical Results

In this section, we compare the numerical performance of the proposed estimator VRS with classical kernel methods and neural network estimators through various density models.

As detailed in Algorithm 2, our approach involves three groups of tuning parameters: m, $\{\ell_j\}_{j=1}^d$, and $\{r_j\}_{j=1}^d$. In all our numerical experiments, the optimal choices for m and $\{\ell_j\}_{j=1}^d$ are determined through cross-validation. To select $\{r_j\}_{j=1}^d$, we apply a popular method in low-rank matrix estimation known as adaptive thresholding. Specifically, for each $j = 1, \ldots, d$, we compute $\{\widehat{\sigma}_{j,k}\}_{k=1}^{\infty}$, the set of singular values of $\widehat{p} \times_j \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j}$. Let τ be a prespecified parameter. We set $r_j = k - 1$ if k is first index such that

$$\widehat{\sigma}_{j,k}^2 < \tau \sum_{\mu=1}^{k-1} \widehat{\sigma}_{j,\mu}^2.$$

This ensures that we can adaptively remove nonzero singular values that are associated with the noise of the data. In our simulations, we set $\tau = 1/50$. Adaptive thresholding is a very popular strategy in the matrix completion literature ([25]) and it has been proven to be empirically robust in many scientific and engineering applications. We use built-in functions provided by the popular Python package scikit-learn to train kernel estimators, and scikit-learn also utilizes cross-validation for tuning parameter selection. For neural networks, we use PyTorch to train various models and make predictions.

We study the numerical performance of Variance-Reduced Sketching (VRS), kernel density estimators (KDE), and neural networks (NN) in various density estimation problems. For neural network estimators, we use two popular density estimation architectures: Masked Autoregressive Flow (MAF) ([170]) and Neural Autoregressive Flows (NAF) ([99]) for comparisons. The details of implementing neural network density estimators are provided in

Section A.6. We measure the estimation accuracy by the relative L_2-error defined as

$$\frac{\|p^* - \widetilde{p}\|_{L_2(\Omega)}}{\|p^*\|_{L_2(\Omega)}},$$

where $\widetilde{p}$ is the density estimator produced by a given estimator. We also compute the standard Kullback–Leibler (KL) divergence to measure the distance between two probability density functions:

$$D_{KL} = \mathbb{E}_{p^*}[\log(p^*/\tilde{p})].$$

Both relative $\mathbf{L}_2$ error and KL divergence are computed via Monte Carlo simulation with a sufficiently large number of test samples.

• **Simulation I: Four-mode Gaussian mixture model.** We consider a four-mode Gaussian mixture model in two dimensions. We generate 20,000 data from the density

$$p^*(\boldsymbol{x}) = \sum_{i=1}^{4} \frac{1}{4} \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}_i|}},$$

where $\boldsymbol{\mu}_1 = \begin{pmatrix} -0.5 \\ -0.5 \end{pmatrix}, \boldsymbol{\mu}_2 = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}, \boldsymbol{\mu}_3 = \begin{pmatrix} -0.5 \\ 0.5 \end{pmatrix}, \boldsymbol{\mu}_4 = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$, and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.25^2 & 0.03^2 \\ 0.03^2 & 0.25^2 \end{pmatrix}$, $\boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{pmatrix} 0.1^2 & -0.05^2 \\ -0.05^2 & 0.1^2 \end{pmatrix}$. We report the relative L_2 error and KL divergence for each method in Table 2.1. To further visualize the performance of the four

	VRS	KDE	MAF	NAF
Relative L_2 Error	0.0721(0.0029)	0.3987(0.0039)	0.2441(0.0411)	0.4617(0.0621)
KL Divergence	0.0142(0.0015)	0.1223(0.0014)	0.0819(0.0161)	0.2356(0.0417)

Table 2.1: Relative L_2 errors and KL divergences for four different methods of the two-dimensional Gaussian mixture model in **Simulation I**. The experiments are repeated 50 times. The average errors are reported and standard deviations are shown in the bracket.

different methods, the true density and the estimated density from each method are plotted

in Figure 2.2. Direct comparison in Figure 2.2 demonstrates VRS provides a relatively better estimate for the true density.

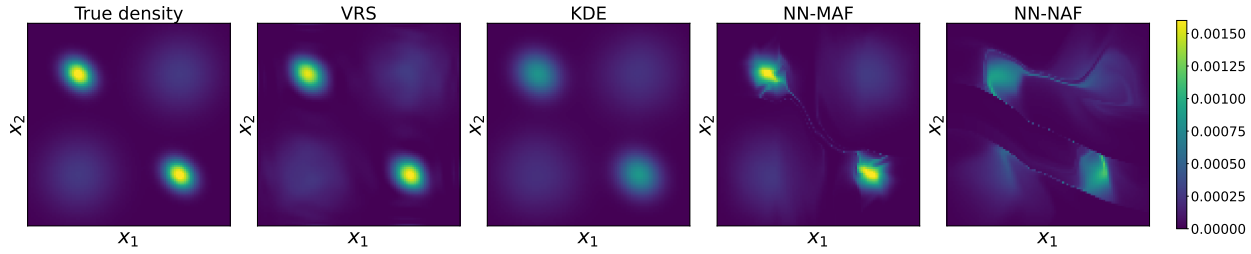


Figure 2.2: Density functions from the two-dimensional Gaussian mixture model in **Simulation I**. From left to right are the ground truth density, estimates from VRS, KDE, MAF, and NAF, respectively. The values in the colorbar on the right represent function values.

• **Simulation II: Two-dimensional Gaussian mixture model.** In the first simulation, we use a two-dimensional two-mode Gaussian mixture model to numerically compare VRS, KDE and neural network estimators. We generate 1,000 samples from the following density function

$$p^*(\boldsymbol{x}) = 0.4\frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right)}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}_1|}} + 0.6\frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)\right)}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}_2|}},$$

where $\boldsymbol{\mu}_1 = \begin{pmatrix} -0.35 \\ -0.35 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{pmatrix} 0.25^2 & -0.03^2 \\ -0.03^2 & 0.25^2 \end{pmatrix}, \boldsymbol{\mu}_2 = \begin{pmatrix} 0.35 \\ 0.35 \end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.35^2 & 0.1^2 \\ 0.1^2 & 0.35^2 \end{pmatrix}$.
The relative L_2 error and KL divergence for each method are reported in Table 2.2. As demonstrated in this example, VRS achieves decent accuracy in the classical low-dimensional setting.

To further visualize the performance of the four different methods, the true density and the estimated density from each method are plotted in Figure 2.3. Direct comparison in Figure 2.3 demonstrates that VRS provides a relatively better estimator for the true density.

	VRS	KDE	NN-MAF	NN-NAF
relative L_2 error	0.1270(0.0054)	0.1636(0.0068)	0.2225(0.0135)	0.3265(0.0103)
KL divergence	0.0092(0.0033)	0.0488(0.0029)	0.0785(0.0134)	0.0983(0.0098)

Table 2.2: Relative L_2 errors and KL divergences for four different methods of the two-dimensional Gaussian mixture model in **Simulation II**. The experiments are repeated 50 times. The average errors are reported and standard deviations are shown in the bracket.

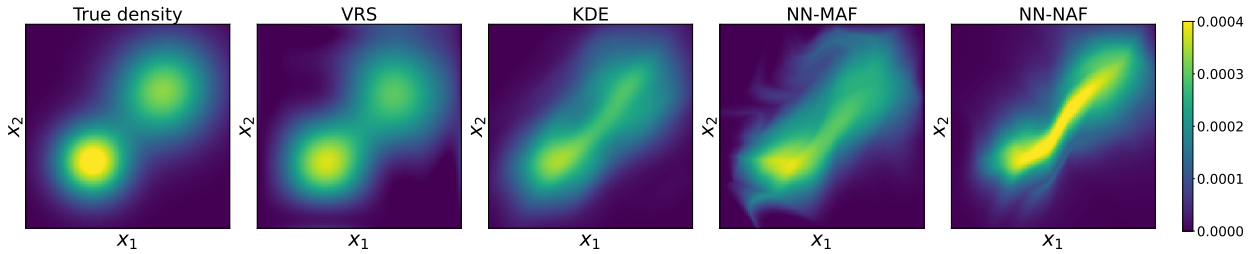


Figure 2.3: Density functions from the two-dimensional Gaussian mixture model in **Simulation II**. From left to right: the ground truth density, and estimations from VRS, KDE, MAF, and NAF. The values in the colorbar on the right represent function values.

• **Simulation III: Gaussian mixture in 30 dimensions.** We consider a higher dimensional density model in this simulation. Specifically, we generate 10^5 data points from the 30-dimensional Gaussian mixture density function

$$\begin{aligned}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} &\overset{\text{i.i.d.}}{\sim} \frac{1}{2}\mathcal{N}\left(\begin{pmatrix} -0.5 \\ -0.5 \\ -0.5 \end{pmatrix}, \begin{pmatrix} 0.1^2 & 0.06^2 & 0 \\ 0.06^2 & 0.1^2 & 0 \\ 0 & 0 & 0.1^2 \end{pmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}, 0.1^2 I_{3\times 3}\right), \\
x_4, x_5 &\overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.04), \\
x_6, \ldots, x_{30} &\overset{\text{i.i.d.}}{\sim} \frac{1}{2}\mathcal{N}(-0.4, 0.3^2) + \frac{1}{2}\mathcal{N}(0.4, 0.3^2).
\end{aligned}$$

The experiments are repeated 50 times. Since computing high-dimensional L_2 errors is *NP-hard*, we only report the averaged KL divergence for performance evaluation. Table 2.3 showcases that VRS outperforms kernel and neural network estimators with a remarkable margin. To further visualize the performance of the four different methods, we provide visualization of a few estimated marginal densities and compare them with the ground truth

	VRS	KDE	MAF	NAF
KL divergence	0.0195(0.0056)	4.3823(0.0047)	0.9260(0.0523)	0.1613(0.0823)

Table 2.3: KL divergence for four methods of the 30-dimensional density model in **Simulation III**. The experiments are repeated for 50 times. The average errors are reported and standard deviations are shown in the bracket.

marginal density. Figure 2.4 depicts the comparison of the two-dimensional marginal densities corresponding to (x_1, x_2), (x_4, x_8), and (x_{10}, x_{20}), respectively. Direct comparison in Figure 2.4 demonstrates VRS provides a relatively better fit for the true density.

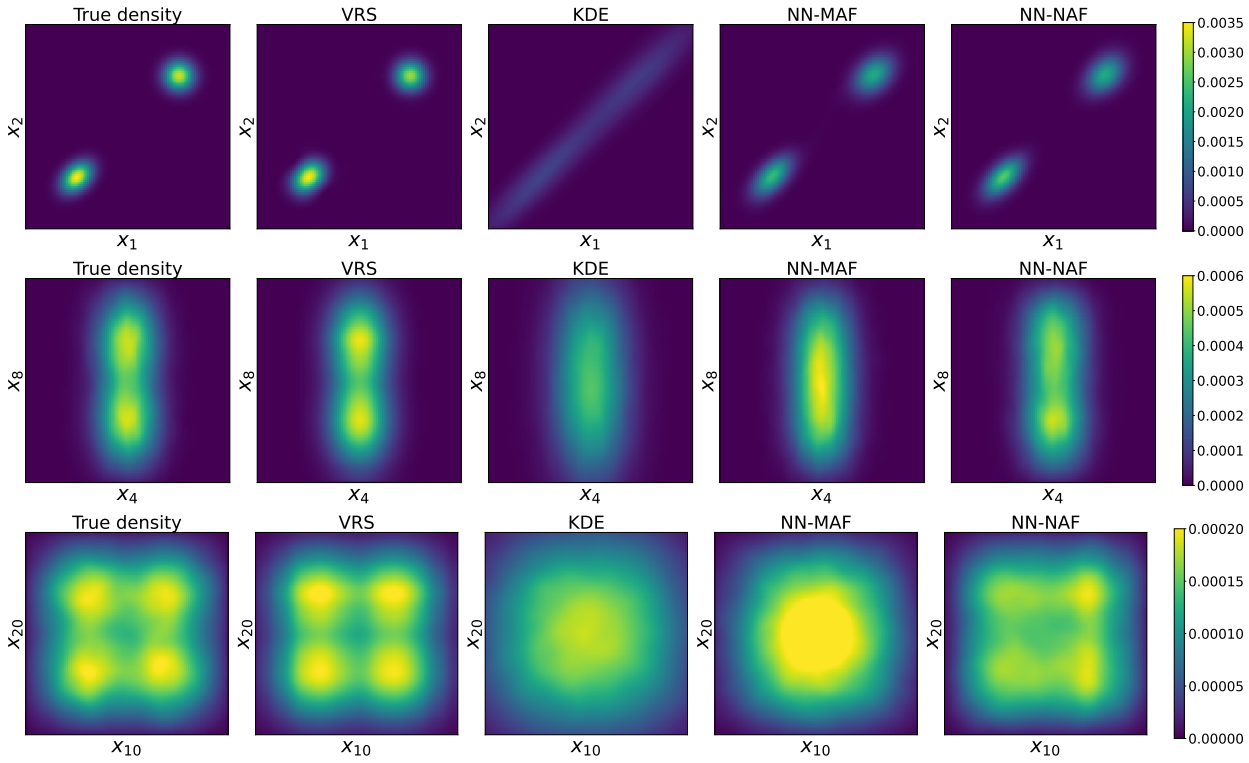


Figure 2.4: Marginal densities from the 30-dimensional Gaussian mixture model in **Simulation III**. From left to right: the ground truth density, and estimations from VRS, KDE, MAF, and NAF. From top to bottom: two-dimensional marginal densities corresponding to (x_1, x_2), (x_4, x_8), and (x_{10}, x_{20}). The values in the colorbar on the right represent function values.

• **Simulation IV: The Ginzburg-Landau model.** Ginzburg-Landau theory is widely used to model the microscopic behavior of superconductors. The Ginzburg-Landau density

has the following expression

$$p^*(x_1,\ldots,x_d) \propto \exp\left(-\beta\left\{\sum_{j=0}^{d}\frac{\lambda}{2}(\frac{x_j-x_{j+1}}{h})^2+\sum_{j=1}^{d}\frac{1}{4\lambda}(x_j^2-1)^2\right\}\right), \tag{2.27}$$

where $x_0 = x_{d+1} = 0$. We sample data from the Ginzburg-Landau density with coefficient $\beta = 1/8, \lambda = 0.02, h = 1/(d+1)$ using the Metropolis-Hastings sampling algorithm. This type of density concentrates on two centers $(+1,+1,\cdots,+1)$ and $(-1,-1,\cdots,-1)$, and all the coordinates $(x_1,\ldots,x_d)$ are correlated in a non-trivial way due to the interaction term $\exp\big(-\beta\sum_{j=0}^{d}\frac{\lambda}{2}(\frac{x_j-x_{j+1}}{h})^2\big)$ in the density function. We consider two sets of experiments for the Ginzburg-Landau density model. In the first set of experiments, we fix $d = 10$ and change the sample size N from 1×10^5 to 5×10^5. In the second set of experiments, we keep the sample size N at 1×10^5 and vary d from 2 to 10. We summarize the averaged relative L_2-error for each method in Figure 2.5. Furthermore in Figure 2.6, we visualize several two-dimensional marginal densities estimated by the four different methods with sample size 1×10^5. Direct comparison showcases that our VRS method recovers these marginal densities with decent accuracy.

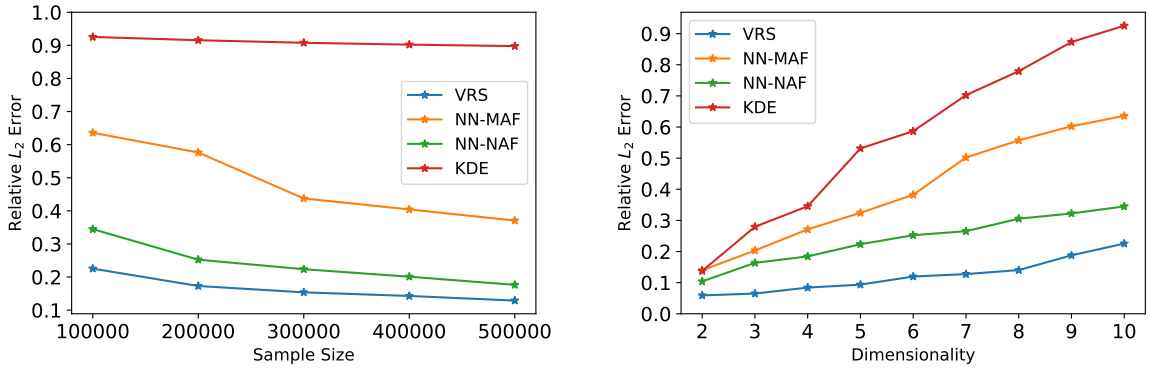


Figure 2.5: The plot on the left corresponds to **Simulation IV** with $d = 10$ and N varying from 1×10^5 to 5×10^5 ; the plot on the right corresponds to **Simulation IV** with N being 1×10^5 and d varying from 2 to 10.

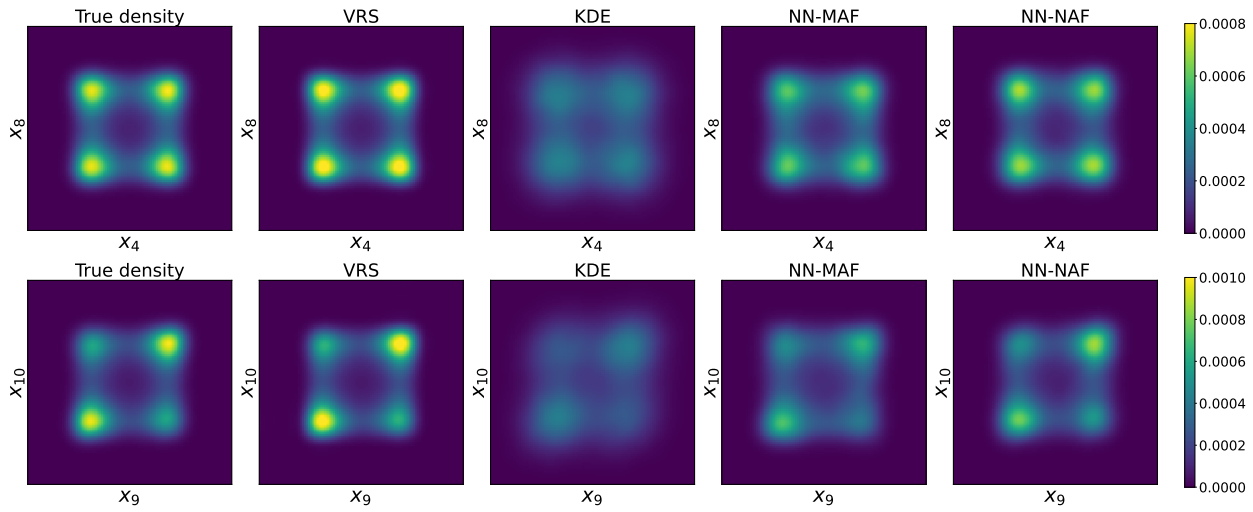


Figure 2.6: Marginal densities from the 10-dimensional Ginzburg-Landau model in **Simulation IV**. From left to right: the ground truth density, and estimations from VRS, KDE, MAF, and NAF. From top to bottom: two-dimensional marginal densities corresponding to (x_4, x_8) and (x_9, x_{10}). The values in the colorbar on the right represent the density function values.

• **Real data example.** We analyze the density estimation for the Portugal wine quality dataset from the UCI Machine Learning Repository. This dataset contains 6497 samples of red and white wines, along with 8 continuous variables: volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, density, sulphates, and alcohol. To provide a comprehensive comparison between different methods, we estimate the joint density of the first d variables in this dataset, allowing d to vary from 2 to 8. For instance, $d = 2$ corresponds to the joint density of volatile acidity and citric acid. Since the true density is unknown, we randomly split the dataset into a 90% training set and a 10% test set, and evaluate the performance of various approaches using the averaged log-likelihood of the test data. The averaged log-likelihood is defined as follows: let $\widetilde{p}$ be the density estimator based on the training data. The averaged log-likelihood of the test data $\{Z_i\}_{i=1}^{N_{\text{test}}}$ is

$$\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \log\{\widetilde{p}(Z_i)\}.$$

The numerical performance of VRS, NN, and KDE is summarized in Figure 2.7. Notably, VRS achieves the highest averaged log-likelihood values, indicating its superior numerical performance.

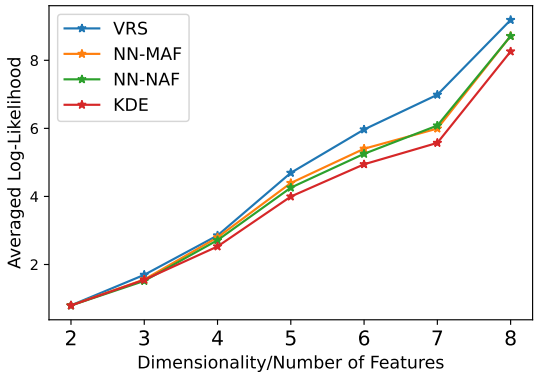


Figure 2.7: Density estimation for the Portugal wine quality dataset with VRS, KDE, and neural network estimators.

2.6 Conclusion

In this paper, we develop a comprehensive framework Variance-Reduced Sketching (VRS) for nonparametric density estimation problems in higher dimensions. Our approach leverages the concept of sketching from numerical linear algebra to address the curse of dimensionality in function spaces. Our method treats multivariable functions as infinite-dimensional matrices or tensors and the selection of sketching is specifically tailored to the regularity of the estimated function. This design takes the variance of random samples in nonparametric problems into consideration, intended to reduce the curse of dimensionality in density estimation problems. Extensive simulated experiments and real data examples demonstrate that our sketching-based method substantially outperforms both neural network estimators and classical kernel density methods in terms of numerical performance.

CHAPTER 3

TENSOR DENSITY ESTIMATOR VIA CONVOLUTION-DECONVOLUTION

3.1 Introduction

Density estimation is a fundamental problem in data science with wide-ranging applications in fields such as science [43, 33], engineering [29], and economics [243]. More precisely, given N data points $\{x^{(i)}\}_{i=1}^{N} \sim p^*$, where each $x^{(i)}$ is an independent and identically distributed (i.i.d.) sample from an unknown ground truth distribution $p^* : \mathbb{R}^d \to \mathbb{R}$, the goal is to estimate p^* based on empirical distribution

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x) \tag{3.1}$$

where $\delta_{x^{(i)}}(x)$ is the Dirac delta function centered at $x^{(i)}$. In general, one cannot use the empirical distribution as an estimator of the density p^*, due to the lack of absolutely continuity or exponentially large variance.

In this work, we introduce a novel framework for high-dimensional density estimation entirely based on linear algebra. Our goal is to achieve overall computational complexity that is linear in dimensionality. This is accomplished by viewing the density estimation process as applying a low-dimensional orthogonal projector in function space onto the noisy $\hat{p}$. Due to the dimension reduction, the exponentially large variance in $\hat{p}$ is suppressed. Additionally, we leverage tensor-train (TT) algebra to perform this projection with linear computational complexity in terms of both dimensionality and sample size. This approach also yields a final density estimator in the form of a tensor train. Furthermore, we theoretically show that, in addition to achieving linear scaling in computational complexity, the estimation error also enjoys linear scaling under certain mild regularity assumptions. Finally, we validate our

method on Boltzmann sampling and real-world generative-modeling tasks, demonstrating performance that matches or exceeds leading deep-learning approaches. A representative result appears in Figure 3.1, with full details in Section 3.9.

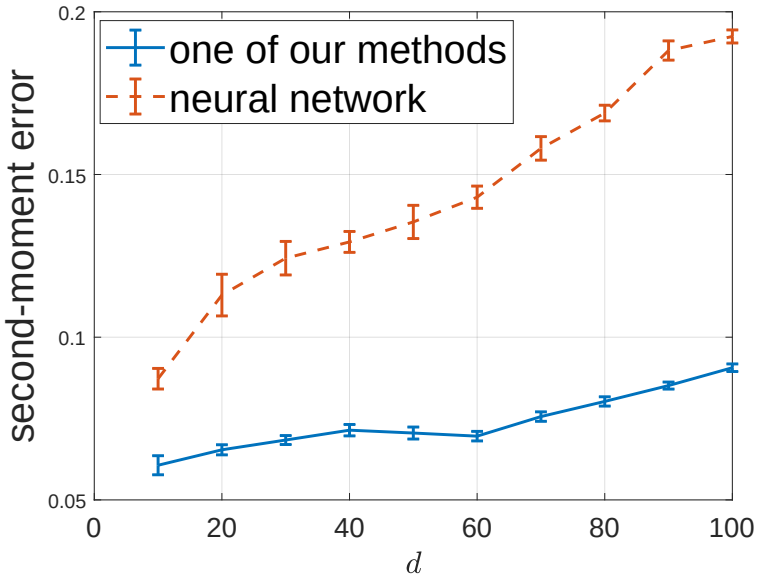


Figure 3.1: Density estimation for data sampled from a Boltzmann distribution. Figure visualizes the second-moment error for different dimensionality d. We compare the result of one of our proposed tensor-train-based methods called TT-SVD-kn (see Section 3.4) with that of one neural network approach (Masked Autoregressive Flow).

Our method is based on a crucial assumption about the true density:

$$p^*(x) = \mu(x)\left(1 + \sum_{j_1=1}^{d}\sum_{l_1=2}^{n} b_{l_1}^{j_1}\phi_{l_1}^{j_1}(x_{j_1}) + \sum_{(j_1,j_2)\in\binom{d}{2}}\sum_{l_1,l_2=2}^{n} b_{l_1 l_2}^{j_1 j_2}\phi_{l_1}^{j_1}(x_{j_1})\phi_{l_2}^{j_2}(x_{j_2})\right), \quad (3.2)$$

where $\mu(x) = \mu_1(x_1)\cdots\mu_d(x_d)$ is a mean-field density, i.e., a separable density. $\{\phi_l^j\}_{l=1}^n$ are orthogonal basis functions with respect to μ_j and $\phi_1^j \equiv 1$, i.e.,

$$\int \phi_{l_1}^j(x_j)\phi_{l_2}^j(x_j)\mu_j(dx_j) = \delta_{l_1,l_2}, \quad (3.3)$$

and $b_{l_1}^{j_1}, b_{l_1 l_2}^{j_1 j_2}$ are the corresponding coefficients. The indices $(j_1, j_2) \in \binom{d}{2}$ denote all unordered pairs of distinct dimensions. In other words, we view p^* as a perturbative expansion

around a mean-field density μ, where each term in the perturbation series involves only a few variables (up to two variables in (3.2)). Although it is straightforward to generalize (3.2) to an expansion that includes higher-order interactions involving more than two variables with decaying coefficients, for clarity of presentation, we use (3.2) as our default model. Such an expansion is common in perturbation and cluster expansion theory in statistical mechanics.

Based on this setting, we discuss several estimators with increasing levels of sophistication. The first approach to estimate p^* is based on a simple and natural subspace projection:

$$\tilde{p} = \mu \odot \left(\Phi \mathrm{diag}(w_K)\Phi^T \hat{p}\right). \tag{3.4}$$

For notational simplicity, we treat all functions and operators above as discrete matrices and vectors: we regard the concatenation of basis functions

$$\Phi := [\Phi_l(x)]_{x,l}$$

as a matrix with rows indexed by x and columns indexed by l. The operation $\Phi^T\hat{p}$ corresponds to convolution between basis functions and $\hat{p}$, i.e., summing or integrating out the variable x. The vector w_K, indexed by l, contains the weights associated with the basis functions and depends on a chosen positive integer K. $\mathrm{diag}(\cdot)$ is the operator producing diagonal matrix with the elements of the input vector on the main diagonal. The mean-field μ is also treated as a vector, indexed by x, and $\odot$ denotes the Hadamard (entrywise) product. Intuitively, the operator $\mu \odot (\Phi \mathrm{diag}(w_K)\Phi^T \cdot)$ resembles a projector that projects $\hat{p}$ onto a function space that only involves a small number of variables determined by the chosen integer chosen integer K. More precisely, let the complete (up to a certain numerical precision) set of k-variable basis functions be defined as:

$$\mathcal{B}_{k,d} = \left\{ \phi_{l_1}^{j_1}(x_{j_1}) \cdots \phi_{l_k}^{j_k}(x_{j_k}) \mid 1 \leqslant j_1 < \cdots < j_k \leqslant d \quad \text{and} \quad 2 \leqslant l_1, \cdots, l_k \leqslant n \right\}. \tag{3.5}$$

which we refer to as the *k-cluster basis* throughout the paper. In particular, $\mathcal{B}_{0,d} = \{1\}$ only contains the constant basis. Furthermore, we say a function is *k-cluster* if it can be expanded by up to k-cluster basis. For example, the perturbed part of p^* in (3.2) is 2-cluster. Based on these definitions, Φ can be formulated as the concatenation of the k-cluster basis across different k:

$$\Phi(x) := \left[\mathcal{B}_{k,d}(x)\right]_{k=0}^{d}. \tag{3.6}$$

Furthermore, we define the weight vector w_K as:

$$w_K(j) := \begin{cases} 1 & j \in \binom{d}{K}, \\ 0 & \text{otherwise}, \end{cases} \tag{3.7}$$

which is an indicator vector that selects the 0- to K-cluster basis. With these definitions, one can verify that if p^* takes the form of (3.2), and if the univariate basis functions $\{\phi_l^j\}_{l=1}^n$ satisfy the orthogonality condition (3.3), then we can recover

$$p^* = \mu \odot \left(\Phi \mathrm{diag}(w_K) \Phi^T p^*\right) \tag{3.8}$$

as long as $K \geqslant 2$. This fact motivates the proposed density estimator in (3.4). Moreover, unlike $\hat{p}$, whose variance scales as $\mathrm{O}(n^d)$, Lemma 1 shows that the variance of $\tilde{p}$ in (3.4) is significantly reduced, scaling as polynomially dependent term $\mathrm{O}(d^K)$ due to the projection onto a lower-dimensional function space. This substantial reduction in variance is a key advantage of our method.

While the estimator in (3.4) is intuitive, it is possible to further reduce the variance by imposing additional assumptions on p^*. Specifically, we assume that the low-order cluster coefficients $\mathrm{diag}(w_K)\Phi^T p^*$ can be represented as a low-rank tensor train. The details of such a representation are provided in the following Section. Under this assumption, our

estimation procedure can be modified as:

$$\tilde{p} = \mu \odot \left(\Phi \left(\texttt{TT-compress} \left(\text{diag}(w_K) \Phi^T \hat{p} \right) \right) \right), \tag{3.9}$$

where we apply certain tensor-train compression algorithm, denoted as `TT-compress`, to the tensor $\text{diag}(w_K)\Phi^T\hat{p}$. The singular value thresholding involved in the compression procedure (`TT-compress`) can further suppress noise in $\text{diag}(w_K)\Phi^T\hat{p}$, leading to a more accurate density estimator.

While the estimator in (3.9) is expected to achieve a lower variance, a major drawback is that the projection $\text{diag}(w_K)\Phi^T\hat{p}$ onto the cluster basis up to order K has a computational complexity of $\text{O}(d^K n^K)$, corresponding to the number of cluster basis functions. This scaling quickly becomes prohibitive as d and K increase. To address this computational bottleneck, we propose a final estimator, which can be viewed as a more efficient variant of (3.9):

$$\tilde{p} = \mu \odot \left(\Phi \text{diag}(\tilde{w}_\alpha)^{-1} \left(\texttt{TT-compress} \left(\text{diag}(\tilde{w}_\alpha) \Phi^T \hat{p} \right) \right) \right), \tag{3.10}$$

where $\tilde{w}_\alpha \in \mathbb{R}^{n^d}$ represents a soft down-weighting of the cluster basis. This approach differs from using w_K, which applies a hard threshold that entirely discards cluster bases involving more than K variables. A key advantage of using such $\tilde{w}_\alpha$ is that it can be designed as a rank-1 tensor, allowing the sequence of operations in (3.10) to be executed with a computational complexity of only $\text{O}(dn)$. This drastically reduces the cost compared to the $\text{O}(d^K n^K)$ complexity in (3.9). Furthermore, the final density estimator $\tilde{p}$ naturally takes the form of a tensor train, allowing essential tasks such as moment computation, sample generation, and pointwise evaluation of the density to scale linearly in d. In Section 3.2.3, we present a simple example demonstrating that the estimator in (3.10) yields accurate results comparable to those of (3.9), thereby illustrating their similar compression capabilities. The study of (3.10) will be the focus of this paper.

Related work

We survey several lines of works that are most closely related to the proposed method.

Density estimation literature. Classical density estimation methods such as histograms [193, 195], kernel density estimators [172, 45] and nearest neighbor density estimates [146] are known for their statistical stability and numerical robustness in low-dimensional cases. But they often struggle with the curse of dimensionality, even in moderately high-dimensional spaces. More recently, deep generative modeling has emerged as a powerful tool for generating new samples, particularly in the context of image, based on neural network architecture. This approach includes energy-based models [114, 60, 203] generative adversarial networks (GANs) [71, 139], normalizing flows [52, 184, 53], and autoregressive models [68, 217, 170]. While the numerical performance for these modern methods is impressive, statistical guarantees and non-convex optimization convergence analysis remain challenging areas of research. Furthermore, for some of these models (e.g. energy based model), generating i.i.d. samples can still be expensive. For methods like GANs, while they can generate samples efficiently, they do not provide a way to evaluate the value of a density at any given point. In short, it is difficult for a method to enjoy all the desirable properties such as easy to train, easy to generate samples, can support evaluation of density values, and can be estimated without the curse of dimensionality.

Tensor train from partial sensing of a function. Tensor train, also known as the matrix product state (MPS) ([233, 177]) in the physics literature, has recently been used when to uncover a function, when only limited number of observations of this function are given. Methods for this include TT-cross [166], DMRG-cross [190], TT completion [136, 201], and others [209, 122, 198]. It has broad applications in high-dimensional scientific computing problems [121, 12] and machine learning [208, 101, 196].

Tensor train for density estimation. In a density estimation setting, we are not

given observed entries of a density function, but just an empirical distribution of it. The key challenge lies in the randomness of the observed object, i.e. the empirical histogram, which contains an exponentially large variance in terms of the dimensionality. Therefore, reducing variance in a high-dimensional setting is crucial to design effective algorithms. Recently, work [103] and subsequent studies [210, 175, 37, 211, 111] employ sketching techniques in tensor decomposition step for density estimation problems, which originates from randomized linear algebra literature. However, most existing algorithms in these works do not have a general way to ensure a linear computational complexity when sketching, in terms of dimensionality. Another approach for density estimation with tensors is through optimization-based method. Works [20, 84, 160] implement gradient descent methods and optimize tensor-network representations according to some loss functions. However, the performance strongly depends on the initialization and often suffers from the problem of local minima due to the highly non-convex nature of the optimization landscape. Furthermore, these iterative procedures require multiple iterations over N data points, resulting in higher computational complexity compared to the previous tensor decomposition algorithms.

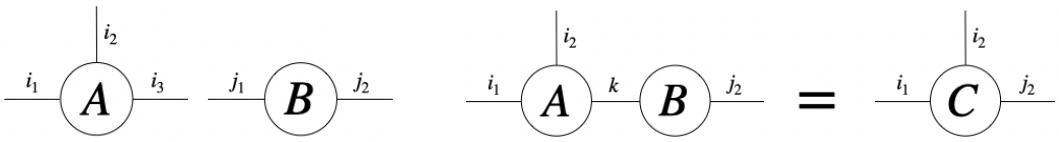
Organization

Here we summarize the organization of the work. In Section 3.2, we present the core ideas behind our proposed "convolution–deconvolution" framework, highlighting its three key steps. The central component is the `TT-compress` algorithm as discussed previously that compresses cluster coefficients into a tensor train. Section 3.3 – 3.6 introduce various `TT-compress` algorithms: Section 3.3 presents a baseline algorithm based on sequential truncated singular value decomposition (SVD), along with its fast implementation for density estimation problems. Section 3.4 includes an algorithm that utilizes the Nyström approximation to further accelerate the algorithm in Section 3.2. In Section 3.5, we study an algorithm that performs SVD efficiently by truncating the elements in the tensor, and then introduce a

further acceleration by employing a hierarchical matrix factorization. Section 3.6 proposes an algorithm utilizing random sketching matrices with low-rank tensor structures. Section 3.7 presents a pre-processing procedure based on principal component analysis (PCA), enabling our compression algorithms to handle very high-dimensional problems. Theoretical analysis for the algorithms is provided in Section 3.8. In Section 3.9, we showcase the comprehensive numerical results to demonstrate the high accuracy and efficiency for our proposed algorithms. Finally, in Section 3.10, we summarize our findings.

Tensor notations and diagrams

We begin with some definitions regarding tensors. A d-dimensional tensor $A \in \mathbb{R}^{n_1 \times n_2 \cdots \times n_d}$ is a collection of entries denoted by $A(i_1, i_2, \cdots, i_d)$ with $1 \leqslant i_j \leqslant n_j$ for $1 \leqslant j \leqslant d$. Diagrammatically, a tensor can be denoted as a node whose dimensionality is indicated by the number of the "legs" of the node. In Figure 3.2(a), the diagrams for a three-dimensional tensor A and a two-dimensional matrix B are plotted as an example. We use i_1, i_2, i_3 to denote the three legs in tensor A. We may also use a rectangle to denote a tensor when it has many legs in this paper.



(a) Tensor diagrams of 3-dimensional A and 2-dimensional B.

(b) Tensor diagram of tensor contraction.

Figure 3.2: Basic tensor diagrams and operations.

Based on the definition of tensor, we list several tensor operations that will be used frequently in the paper:

1. **"$\circ$" contraction.** We use $\circ$ to represent the tensor cores contraction $(A \circ B)(i_1, i_2, j_2) = \sum_k A(i_1, i_2, k) B(k, j_2)$. Such operation is denoted as diagram in Figure 3.2 (b) which

combines the last leg of A and the first leg of B. The resulting $C = A \circ B$ is a three-dimensional tensor.

2. **Unfolding matrix.** Given a d-dimensional tensor with entries $A(i_1, i_2, \cdots, i_d)$, if we group the first j indices into a multi-index $i_{1:j}$ and group the rest into another multi-index $i_{j+1:d}$, we will obtain a *j-th unfolding matrix* $A^{(j)} \in \mathbb{R}^{(n_1 \cdots n_j) \times (n_{j+1} \cdots n_d)}$ (also called j-th unfolding of A) satisfying

$$A^{(j)}(i_{1:j}, i_{j+1:d}) = A(i_1, i_2, \cdots, i_d). \tag{3.11}$$

In particular, $A^{(0)} \in \mathbb{R}^{1 \times n_1 \cdots n_d}$ and $A^{(d)} \in \mathbb{R}^{n_1 \cdots n_d \times 1}$ are respectively the row and column vectors reshaped from the tensor.

In general, the memory cost of a high-dimensional tensor suffers from curse of dimensionality. Tensor-train representation is a way to decompose the whole exponential large d-dimensional tensor with underlying low-rank structure into d number of components:

$$A = G_1 \circ G_2 \circ \cdots \circ G_d \in \mathbb{R}^{n_1 \times \cdots \times n_d} \tag{3.12}$$

where $G_1 \in \mathbb{R}^{n_1 \times r_1}, G_2 \in \mathbb{R}^{r_1 \times n_2 \times r_2}, \cdots, G_{d-1} \in \mathbb{R}^{r_{d-2} \times n_{d-1} \times r_{d-1}}, G_d \in \mathbb{R}^{r_{d-1} \times n_d}$ are called tensor cores and $\{r_j\}_{j=1}^{d-1}$ are the ranks. The TT representation can be visualized as the left panel of Figure 3.3. Throughout the paper, we denote the largest rank by

$$r = \max_j r_j.$$

Then the memory cost for storing A in TT form is at $\mathrm{O}(dnr^2)$, which has no curse of dimensionality if r is small. In addition, a TT can also be used to represent a continuous

function $f(x_1, \cdots, x_d)$ formulated as a tensor product basis function expansion:

$$f(x_1, x_2, \cdots, x_d) = \sum_{l_1, l_2, \cdots, l_d = 1}^{n} A(l_1, l_2, \cdots, l_d)\phi_{l_1}(x_1)\phi_{l_2}(x_2)\cdots\phi_{l_d}(x_d)$$

where the coefficient A takes the TT format as (3.12) . Such functional tensor-train representation is illustrated in the right panel of Figure 3.3.

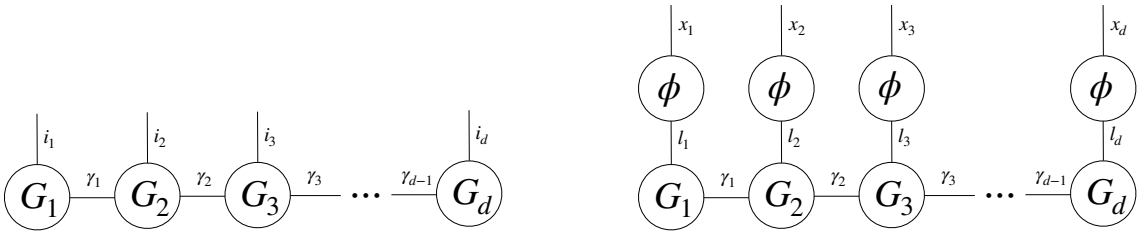


Figure 3.3: Diagram for tensor-train representation. Left: discrete case. Right: continuous case.

3.2 Main Approach

In this section, we detail the framework of our main approach. In order to perform (3.10), we perform the following three steps:

1. **Convolution.** Project the empirical distribution $\hat{p}$ to "coefficients" $\hat{c}$ under certain basis expansion with $\mathrm{O}(dN)$ computational cost:

$$\hat{c} = \mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p} = \mathrm{diag}(\tilde{w}_\alpha)\int \Phi(x)^T\hat{p}(x)dx. \tag{3.13}$$

2. **Tensor compression.** Approximate $\hat{c}$ as a TT $\tilde{c} = \texttt{TT-compress}(\hat{c})$. This step can be done by several proposed algorithms listed in Table 3.1. Notably, several of these algorithms achieve $\mathrm{O}(dN)$ complexity. These complexity estimates will be further validated through numerical experiments in Section 3.9.

Algorithm	Corresponding section	Computational complexity
TT-SVD	§3.3	$\mathrm{O}(\min(dn^{d+1}, dN^2n))$
TT-SVD-kn	§3.4	$\mathrm{O}(dNn)$
TT-SVD-c (K-cluster)	§3.5	$\mathrm{O}(d^{K+1}Nn^{K+1})$
TT-SVD-c (hierarchical)	§3.5.1	$\mathrm{O}(d\log(d)Nn)$
TT-rSVD-t	§3.6	$\mathrm{O}(dNn)$

Table 3.1: Computational complexity of proposed `TT-compress`.

3. **Deconvolution.** Get the desired density estimator based on the TT coefficient $\tilde{c}$ with $\mathrm{O}(dN)$ computational cost:

$$\tilde{p} = \mu \odot (\Phi \mathrm{diag}(\tilde{w}_\alpha)^{-1}\tilde{c}). \tag{3.14}$$

Here $\tilde{p}$ is also a TT, if $\Phi\mathrm{diag}(\tilde{w}_\alpha)^{-1}$ has an appropriate low-rank tensor structure.

Overall, the computational cost of our approach (with proper choices of `TT-compress`) can achieve computational complexity that scales linearly with both the dimensionality d and sample size N, demonstrating the high efficiency. In the rest of this section, we will further elaborate convolution (Step 1) and deconvolution (Step 3). We defer the details of tensor compression (Step 2) to Section 3.3 – 3.6.

3.2.1 Convolution Step

The key part of this step is the choice of $\tilde{w}_\alpha \in \mathbb{R}^{n^d}$, the soft down-weighting of the cluster basis (3.6). In this work, we choose the weight of any k-cluster basis $\phi_{l_1}^{j_1}(x_{j_1}) \cdots \phi_{l_k}^{j_k}(x_{j_k})$ to be

$$\alpha^k, \text{ where parameter } \alpha \in (0, 1]. \tag{3.15}$$

In this way, the coefficient tensor $\hat{c} = \mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p}$ has entries

$$\hat{c}(l_1, l_2, \cdots, l_d) = \frac{1}{N}\sum_{i=1}^{N} \alpha^{\|l-\mathbf{1}\|_0} \prod_{j=1}^{d} \phi_{l_j}^{j}(x_j^{(i)}).$$

where $\|\cdot\|_0$ is the zero norm counting the number of non-zero entries of a vector, and $\mathbf{1}$ is the all-one d-dimensional vector. $\|l-\mathbf{1}\|_0$ is then the number of non-constant univariate basis functions among $\phi_{l_1}^{j_1}(x_{j_1})\cdots\phi_{l_k}^{j_k}(x_{j_k})$. Under such choice, $\mathrm{diag}(\tilde{w}_\alpha)$ can be formulated as a rank-1 tensor operator, allowing the fast convolution (and also fast deconvolution later) with only $\mathrm{O}(dN)$ computational cost. In fact, given samples $\{x^{(i)}\}_{i=1}^N$, $\hat{c}$ is formulated as

$$\hat{c} = \frac{1}{N}\sum_{i=1}^{N} \tilde{\Phi}_1^{(i)} \otimes \tilde{\Phi}_2^{(i)} \otimes \cdots \otimes \tilde{\Phi}_d^{(i)} \tag{3.16}$$

where $\tilde{\Phi}_j^{(i)} = [1, \alpha\phi_2^j(x_j^{(i)}), \cdots, \alpha\phi_n^j(x_j^{(i)})] \in \mathbb{R}^n$. This expression is plotted as tensor diagrams in Figure 3.4. For future use, we introduce

$$\tilde{\Phi}_j \in \mathbb{R}^{N\times n} \tag{3.17}$$

to denote the matrix whose rows are $\tilde{\Phi}_j^{(i)}$.

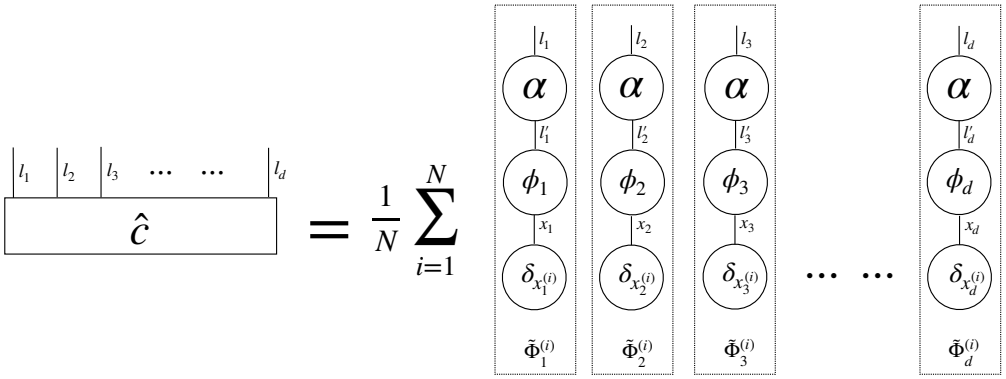


Figure 3.4: Tensor diagram of $\hat{c} = \mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p}$. "$\alpha$" denotes diagonal matrix $\mathrm{diag}(1, \alpha, \cdots, \alpha) \in \mathbb{R}^{n\times n}$; "$\phi_j$" denotes "matrix" $[\phi_l^j(x)]_{x,l}$.

The idea of such choice is to assign smaller weights for higher-order cluster basis to suppress the variance. The parameter α describes the denoising strength — a smaller α produces stronger denoising and thus smaller variance. In particular when $\alpha = 1$, this soft-thresholding method (3.10) reduces to the hard-thresholding (3.9) using all cluster bases. We point out that by choosing sufficiently small α, this soft-thresholding strategy has smaller variance compared to the hard-thresholding approach (3.9). The following lemma considers a toy model to show the variance of the hard-thresholding approach:

Lemma 1. *Suppose* $\{\phi_l\}_{l=1}^n$ *are Legendre polynomials and the ground truth density* p^* *satisfies* (3.8) *with the mean-field component* $\mu \sim \mathrm{Unif}([0,1]^d)$*, the following variance estimation holds*

$$\mathbb{E}\|\Phi \mathrm{diag}(w_K)\Phi^T \hat{p} - \Phi \mathrm{diag}(w_K)\Phi^T p^*\|_{L_2}^2 = \mathrm{O}(\frac{d^K n^{2K}}{N}). \tag{3.18}$$

Proof. Note that the error $\|\mathrm{diag}(w_K)\Phi^T \hat{p} - \mathrm{diag}(w_K)\Phi^T p^*\|_F^2$ has at most $\binom{d}{K} n^K$ terms, where each term follows

$$\left(\int (\prod_{k=1}^K \phi_{l_k}^{j_k}(x_{j_k}))(\hat{p}(x) - p^*(x))dx \right)^2 \tag{3.19}$$

for indices $(j_1, \cdots, j_K) \in \binom{d}{K}$ and $l_1, \cdots, l_K \in \{2, \cdots, n\}$. Since Legendre polynomial satisfies $\|\phi_{l_k}^{j_k}\|_\infty \leqslant \sqrt{n}$, then

$$\mathbb{E}\left(\int (\prod_{k=1}^K \phi_{l_k}^{j_k}(x_{j_k}))(\hat{p}(x) - p^*(x))dx \right)^2 = \frac{1}{N} \int (\prod_{k=1}^K \phi_{l_k}^{j_k}(x_{j_k}))^2 p^*(x)dx \leqslant \frac{n^K}{N} \int p^*(x)dx = \frac{n^K}{N} \tag{3.20}$$

Therefore, the total variance is bounded by

$$\begin{aligned} &\mathbb{E}\|\Phi \mathrm{diag}(w_K)\Phi^T \hat{p} - \Phi \mathrm{diag}(w_K)\Phi^T p^*\|_{L_2}^2 = \mathbb{E}\|\mathrm{diag}(w_K)\Phi^T \hat{p} - \Phi^T p^*\|_F^2 \\ &\leqslant \binom{d}{K} n^K \frac{n^K}{N} = \mathrm{O}(\frac{d^K n^{2K}}{N}). \end{aligned} \tag{3.21}$$

□

On the other hand, the variance of the soft-thresholding approach under the model assumption in Lemma 1 is estimated by

$$\begin{aligned}
&\mathbb{E}\left[\|\Phi\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p} - \Phi\mathrm{diag}(\tilde{w}_\alpha)\Phi^T p^*\|_{L_2}^2\right] \\
&= \mathbb{E}\left[\|\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p} - \mathrm{diag}(\tilde{w}_\alpha)\Phi^T p^*\|_F^2\right] \\
&= \sum_{l_1,\cdots,l_d=1}^{n} \alpha^{2\|l-\mathbf{1}\|_0}\mathbb{E}\left[\int (\hat{p}(x) - p^*(x))\left(\prod_{j=1}^{d}\phi_{l_j}^j(x_j)\right)\mathrm{d}x\right]^2 \\
&= \frac{1}{N}\sum_{l_1,\cdots,l_d=1}^{n} \alpha^{2\|l-\mathbf{1}\|_0}\int\prod_{j=1}^{d}(\phi_{l_j}^j(x_j))^2 p^*(x)\mathrm{d}x \\
&\leqslant \frac{1}{N}\sum_{l_1,\cdots,l_d=1}^{n} \alpha^{2\|l-\mathbf{1}\|_0} n^{\|l-\mathbf{1}\|_0}\int p^*(x)\mathrm{d}x \\
&= \frac{1}{N}\sum_{l_1,\cdots,l_d=1}^{n} (\alpha^2 n)^{\|l-\mathbf{1}\|_0} = \frac{1}{N}(1+(n-1)n\alpha^2)^d
\end{aligned} \tag{3.22}$$

where the inequality follows $\|\phi_{l_j}^j\|_\infty \leqslant \sqrt{n}$ for $l_j \geq 2$ and we use the binomial theorem for the last equality. With this estimation, we conclude that if we set $\alpha = 1$, the variance scales at least $\mathrm{O}(n^d)$, indicating the curse of dimensionality. To avoid this, we choose

$$\alpha = \sqrt{\frac{C}{(n-1)nd}}.$$

for some constant C without d-dependency. Under this choice, (3.22) can be bounded by

$$\mathbb{E}\left[\|\Phi\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p} - \Phi\mathrm{diag}(\tilde{w}_\alpha)\Phi^T p^*\|_{L_2}^2\right] \leqslant \frac{1}{N}(1+\frac{C}{d})^d \leqslant \frac{1}{N}e^C \tag{3.23}$$

The resulting error has no curse of dimensionality. However, we also point out that α should not be chosen to be too small, which might lead to instability in our later operations. This

will be further elaborated in Section 3.2.2.

3.2.2 Deconvolution Step

After we have received the approximated coefficient $\tilde{c}$ in TT format:

$$\tilde{c} = \hat{G}_1 \circ \hat{G}_2 \circ \cdots \circ \hat{G}_d \in \mathbb{R}^{n \times n \times \cdots \times n}.$$

in tensor compression step, the last step then obtains the final density estimator in TT format by implementing the deconvolution

$$\tilde{p} = \mu \odot (\Phi \mathrm{diag}(\tilde{w}_\alpha)^{-1} \tilde{c}). \tag{3.24}$$

The structure of $\Phi \mathrm{diag}(\tilde{w}_\alpha)^{-1}\tilde{c}$ is illustrated in Figure 3.5. Since $\tilde{c}$ is a low-rank tensor train, the complexity of the evaluation on a single point x is only $\mathrm{O}(dn)$.

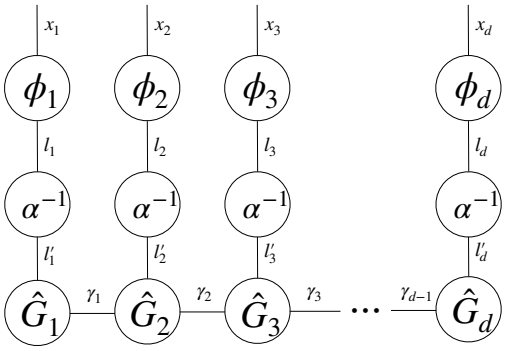


Figure 3.5: Tensor diagram for the final density estimator $\tilde{p}$. "α^{-1}" denotes diagonal matrix $\mathrm{diag}(1, 1/\alpha, \cdots, 1/\alpha) \in \mathbb{R}^{n \times n}$; "$\phi_j$" denotes "matrix" $[\phi_l^j(x)]_{x,l}$.

Remark 1. *In practice, as dimensionality d grows, some elements in $diag(\tilde{w}_\alpha)$ become exponentially small, which potentially lead to instability when computing the inverse. One solution is to implement inverse with given threshold: $\tilde{p} = \Phi(\mathrm{diag}(\alpha) + \lambda I)^{-1}\tilde{c}$. Here λ is a small thresholding constant to avoid instability issues and I is identity matrix. In practice,*

we find that even without this post-procedure, the tensor-train estimator does not suffer from such an instability.

3.2.3 Motivation: Mean-field Example

In this subsection, we walk through the main steps of the proposed approach using a mean-field example of the ground truth density, i.e., $p^* = \mu$ in (3.2). This toy example is used to illustrate why the soft-thresholding strategy in (3.10) effectively projects an empirical density onto the space spanned by low-order cluster bases, providing a similar effect as (3.9). The analysis can be extended to general examples.

For convenience, we first introduce some notations for moments: we define the first-order moment $\hat{q}_j \in \mathbb{R}^{(n-1)\times 1}$ as

$$\hat{q}_j(l_j) = \int \phi^j_{l_j+1}(x_j)\hat{p}(x)\mathrm{d}x, \qquad l_j = 1, \cdots, n-1, \tag{3.25}$$

and the second-order moment $\hat{q}_{j,j'} \in \mathbb{R}^{(n-1)\times(n-1)}$

$$\hat{q}_{j,j'}(l_j, l'_{j'}) = \int \phi^j_{l_j+1}(x_j)\phi^{j'}_{l'_{j'}+1}(x_{j'})\hat{p}(x)\mathrm{d}x, \qquad l_j, l'_{j'} = 1, \cdots, n-1. \tag{3.26}$$

Higher-order moments can be defined following the same logic.

We consider to compress the tensor $\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p}$ into a tensor train using one specific approach called CUR approximation ([147]). Specifically, a given matrix $A \in \mathbb{R}^{m\times n}$ can be approximated by the following rank-r factorization

$$A = CU^+R$$

where $C \in \mathbb{R}^{m\times r}$ and $R \in \mathbb{R}^{r\times n}$ respectively consist of selected r columns and rows. U is the intersection (cross) submatrix formed by the selected rows and columns, and U^+ is

the Moore-Penrose generalized inverse of U. To ensure a good approximation, the columns and rows are typically chosen to maximize the determinant of U. Repeating the procedure for all dimensions, we can receive a tensor-train decomposition eventually. Further details will be discussed in Section 3.4. Based on the notations above, we first unfold the weighted projection tensor $\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p}$ as

$$(\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p})^{(1)} = \begin{bmatrix} 1 & \alpha[\hat{q}_j^{(0)}]_{j=2}^d & \alpha^2[\hat{q}_{j,j'}^{(0)}]_{j,j'=2}^d & \cdots \\ \alpha\hat{q}_1 & \alpha^2[\hat{q}_{1,j}^{(1)}]_{j=2}^d & \alpha^3[\hat{q}_{1,j,j'}^{(1)}]_{j,j'=2}^d & \cdots \end{bmatrix} =: B_1 \tag{3.27}$$

in terms of the order of α, where we use $[\cdot]$ to represent the concatenation of those inside terms through the column. Since the ground truth density p^* is mean-field (rank-1), we choose only $r = 1$ column and row for CUR approximation of $(\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p})^{(1)}$. With α chosen to be sufficiently small, the $(1,1)$-entry is the largest in B_1. Therefore, first row and first column are chosen for the CUR approximation:

$$B_1 \approx \begin{bmatrix} 1 \\ \alpha\hat{q}_1 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & \alpha[\hat{q}_j^{(0)}]_{j=2}^d & \alpha^2[\hat{q}_{j,j'}^{(0)}]_{j,j'=2}^d & \cdots \end{bmatrix}}_{=:B_2} \tag{3.28}$$

The left vector $\begin{bmatrix} 1 \\ \alpha\hat{q}_1 \end{bmatrix}$ is the resulting first TT core (before normalization). Next, we continue to implement the cross approximation on the unfolding of remaining tensor B_2. First row and first column are again chosen following the same logic:

$$B_2 = \begin{bmatrix} 1 & \alpha[\hat{q}_j^{(0)}]_{j=3}^d & \alpha^2[\hat{q}_{j,j'}^{(0)}]_{j,j'=3}^d & \cdots \\ \alpha\hat{q}_2 & \alpha^2[\hat{q}_{2,j}^{(1)}]_{j=3}^d & \alpha^3[\hat{q}_{2,j,j'}^{(1)}]_{j,j'=3}^d & \cdots \end{bmatrix} \approx \begin{bmatrix} 1 \\ \alpha\hat{q}_2 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & \alpha[\hat{q}_j^{(0)}]_{j=3}^d & \alpha^2[\hat{q}_{j,j'}^{(0)}]_{j,j'=3}^d & \cdots \end{bmatrix}}_{=:B_3}. \tag{3.29}$$

We repeat this procedure until decomposing the tensor into the product of d rank-1 cores:

$$\operatorname{diag}(\tilde{w}_\alpha)\Phi^T\hat{p} \approx \begin{bmatrix} 1 \\ \alpha\hat{q}_1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \alpha\hat{q}_2 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ \alpha\hat{q}_d \end{bmatrix} =: \texttt{TT-compress}\left(\operatorname{diag}(\tilde{w}_\alpha)\Phi^T\hat{p}\right). \quad (3.30)$$

Furthermore, we try to estimate the difference between the representation in (3.30) and hard truncation $\texttt{TT-compress}\left(\operatorname{diag}(w_K)\Phi^T\hat{p}\right)$ in (3.9), which are the summation of all k-cluster terms for $k > K$. Since ground truth density p^* is mean-field, the expectation of empirical one-marginal is $\mathbb{E}[\hat{q}_j] = 0$. Besides, variance $\operatorname{Var}[\hat{q}_j] = \mathrm{O}(\frac{n-1}{N})$ for each j due to the orthogonality of $\{\phi_l^j\}_{l=1}^n$ with respect to μ. Therefore, we can have an upper bound of k-cluster, $\alpha^k\|\hat{q}_{j_1} \otimes \cdots \otimes \hat{q}_{j_k}\|_F \leqslant C\alpha^k(\frac{n}{N})^{k/2}$ for some constant $C > 0$. Then compared to the hard-thresholding K-cluster estimator in (3.9), the difference towards the soft-thresholding estimator (3.10) is upper bounded by

$$\begin{aligned} &\| \left(\texttt{TT-compress}\left(\operatorname{diag}(w_K)\Phi^T\hat{p}\right)\right) - \left(\texttt{TT-compress}\left(\operatorname{diag}(\tilde{w}_\alpha)\Phi^T\hat{p}\right)\right) \|_F \\ &\leqslant C\sum_{k=K+1}^{d}\binom{d}{k}\alpha^k(\frac{n}{N})^{k/2} \leqslant C\sum_{k=K+1}^{d}(\frac{\alpha de}{K}\sqrt{\frac{n}{N}})^k \leqslant Cd(\frac{\alpha de}{K}\sqrt{\frac{n}{N}})^{K+1} \end{aligned} \quad (3.31)$$

when $\frac{\alpha de}{K}\sqrt{\frac{n}{N}} < 1$ for sufficient large N. The difference between two estimators is polynomially small with respect to sample size N. This turns out to be a clear example to demonstrate the reliability of soft-thresholding approach (3.10).

3.3 Tensor-train SVD (TT-SVD)

Starting from this section, we focus on Step 2 of the main approach — tensor compression. To this end, we first propose to use the tensor-train SVD (TT-SVD) algorithm. The TT-SVD algorithm is originally proposed in [168], which compresses a high-dimensional tensor into a low-rank tensor train by performing truncated SVD sequentially. For our problem,

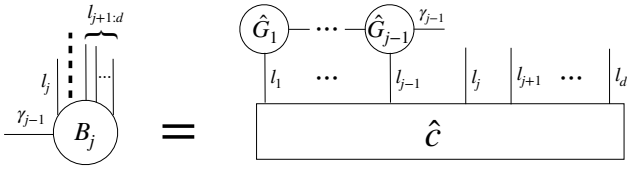
we use TT-SVD to find the approximation $\tilde{c}$ in TT format of $\hat{c}$ depicted in Figure 3.4, which can be obtained following the procedures in Algorithm 3.

Algorithm 3 TT-SVD

INPUT: Tensor $\hat{c} \in \mathbb{R}^{n \times n \times \cdots \times n}$. Target rank $\{r_1, \cdots, r_{d-1}\}$.

1: **for** $j \in \{1, \cdots, d-1\}$ **do**

2: Perform SVD to $B_j \in \mathbb{R}^{nr_{j-1} \times n^{d-j}}$, which is formed by the following tensor contraction



In particular, $B_1 = \hat{c}^{(1)}$ which is the first unfolding matrix of $\hat{c}$ defined in (3.11). With SVD $B_j = U_j \Sigma_j V_j^T$, we obtain the j-th core $\hat{G}_j \in \mathbb{R}^{r_{j-1} \times n \times r_j}$, which is reshaped from the first r_j principle left singular vectors in U_j.

3: **end for**

4: Obtain the last core $\hat{G}_d = B_d \in \mathbb{R}^{r_{d-1} \times n}$.

OUTPUT: Tensor train $\tilde{c} = \hat{G}_1 \circ \hat{G}_2 \circ \cdots \circ \hat{G}_d \in \mathbb{R}^{n \times n \times \cdots \times n}$.

Below, we briefly outline the idea behind this algorithm. When $j = 1$, we want to obtain the first core $\hat{G}_1$ by performing SVD to $\hat{c}^{(1)}$ (which is B_1 in Algorithm 3). Then we let $\hat{G}_1 \in \mathbb{R}^{n \times r_1}$ to be first r_1 principle left singular vectors. The rationale of the first step is to approximate $\hat{c}^{(1)}$ as $\hat{c}^{(1)} \approx \hat{G}_1(\hat{G}_1^T \hat{c}^{(1)})$. In the $j = 2$ step, we want to obtain $\hat{G}_2$, which gives a further approximation to $\hat{G}_1(\hat{G}_1^T \hat{c}^{(1)})$. The way we do this is to factorize $\hat{G}_1^T \hat{c}^{(1)} \in \mathbb{R}^{r_1 \times n^{d-1}}$, by applying SVD to B_2, which is the reshaping of $\hat{G}_1^T \hat{c}^{(1)}$. This completes the $j = 2$ step. Then we do the above procedure recursively till $j = d - 1$. After implementing sequential SVD for $d - 1$ times, the remaining $B_d \in \mathbb{R}^{r_{d-1} \times n}$ is a matrix and we take it as the final core in the tensor-train representation since no more decomposition is required.

Naively, the computational cost from SVD in Algorithm 3 scales exponentially large as $\mathrm{O}(dn^{d+1})$ in terms of dimensionality. However, for the density estimation problem, we can take advantage of the structure of $\hat{c}$ as depicted in Figure 3.4 to achieve a fast implementation of Algorithm 3. We will use the next subsection to elaborate such fast implementation.

3.3.1 Fast Implementation

In this subsection, we present a fast implementation of Algorithm 3 with a complexity of $\mathrm{O}(dN^2nr)$, which scales linearly with d and is therefore well-suited for high-dimensional cases. Notably, this efficient implementation introduces no additional approximations and produces the same output as the vanilla implementation in Algorithm 3.

The key of the fast implementation lies in fast calculations of SVD of B_j in Algorithm 3. Instead of performing SVD to B_j directly to get its left singular vectors as in Line 2 of Algorithm 3, we consider computing the eigenvectors of $B_jB_j^T$, which is a relatively small matrix of size $nr_{j-1} \times nr_{j-1}$. However, one concern is that the cost of forming $B_jB_j^T$ can in principle scale exponentially with dimensionality, since the column size of B_j is n^{d-j} as indicated by multiple legs between the two nodes in Figure 3.6. Fortunately, by using the structure of $\hat{c}$ as shown in Figure 3.4, one can form $B_jB_j^T$ much more efficiently. This is shown in Line 2 of Algorithm 4, which summarizes the TT-SVD procedure that is implemented in a computationally feasible manner.

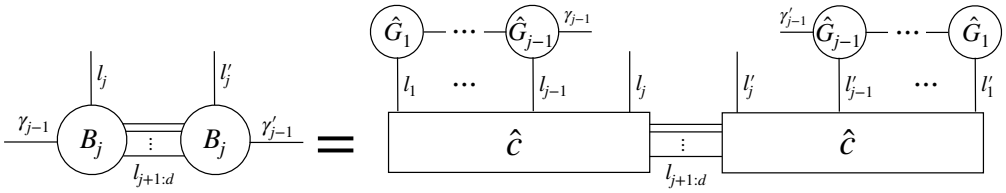


Figure 3.6: Tensor components for forming $B_jB_j^T$.

In Algorithm 4, the main computation lies in forming D_j and E_j in Line 2. A naive way is to compute them independently, leading to $\mathrm{O}(dN^2)$ complexity. However, D_j and

E_j can be computed in a recursive way efficiently, where computations can be reused. More specifically, one can precompute E_j by

$$E_j = E_{j+1} \odot (\tilde{\Phi}_j \tilde{\Phi}_j^T) \tag{3.32}$$

starting from $E_d = I_{N\times N}$ which is $N \times N$ the identity matrix. D_j is computed recursively based on core $\hat{G}_{j-1}$ starting from $D_1 = \tilde{\Phi}_1^T$ according to Figure 3.7. The last core $\hat{G}_d$ is assigned as the matrix B_d in algorithm 3, which is equal to the average of $D_d^{(i)}$ over sample index i. Therefore, we compute $\hat{G}_d$ as the average of $D_d^{(i)}$ in Algorithm 4. In this way, the computational cost of E_j, D_j and $B_j B_j^T$ is $\mathrm{O}(N^2 nr)$ which has no dependency on dimensionality. Afterwards, one can solve the core by eigen-decomposition of $B_j B_j^T \in \mathbb{R}^{nr_{j-1}\times nr_{j-1}}$ with a minor cost as $\mathrm{O}((nr)^3)$. Overall, the computational cost of this fast implementation is $\mathrm{O}(dN^2 nr)$.

While this fast implementation significantly reduces the computational complexity from exponential to linear in d, we point out that such implementation still requires $\mathrm{O}(N^2)$ computations to form each E_j, which can make it challenging with a large number of samples even in low dimensional cases. In Section 3.4, we will further improve Algorithm 4 using kernel Nyström approximation, which will bring computational cost to $\mathrm{O}(dN)$ only.

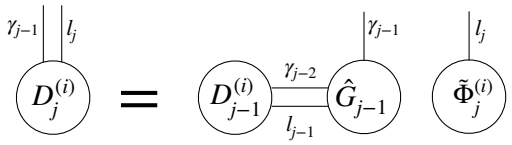


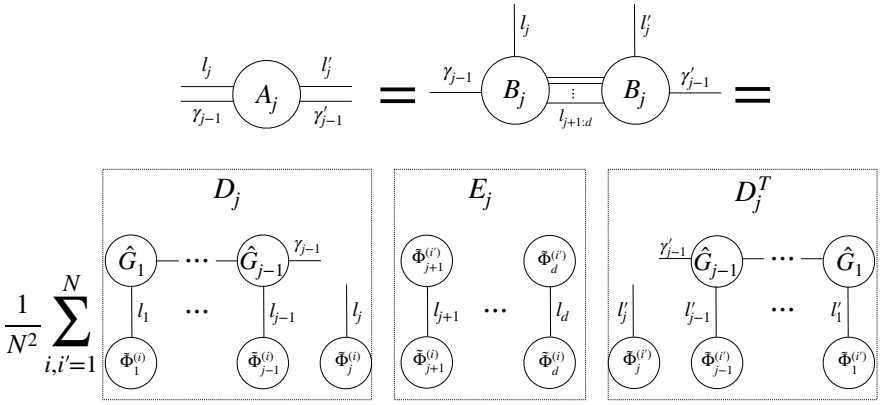
Figure 3.7: Tensor diagram illustrating the recursive calculation of $D_j^{(i)}$, which is the i-th column of the matrix D_j in Algorithm 4.

Algorithm 4 Fast implementation of TT-SVD

INPUT: Samples $\{x^{(i)}\}_{i=1}^N$. Target rank $\{r_1, \cdots, r_{d-1}\}$.

1: **for** $j \in \{1, \cdots, d-1\}$ **do**

2: Perform eigen-decomposition to $A_j := B_j B_j^T \in \mathbb{R}^{nr_{j-1} \times nr_{j-1}}$, which is computed by $A_j = \frac{1}{N^2} D_j E_j D_j^T$ where $D_j \in \mathbb{R}^{nr_{j-1} \times N}$ and $E_j \in \mathbb{R}^{N \times N}$ are the resulting matrices by contraction in the boxes of the below figure:



In particular, $D_1 = \tilde{\Phi}_1^T$ defined in (3.17). Here D_j and E_j should be computed recursively to reuse calculations. With eigen-decomposition $A_j = U_j \Lambda_j U_j^T$, we obtain the j-th core $\hat{G}_j \in \mathbb{R}^{r_{j-1} \times n \times r_j}$, which is reshaped from the first r_j principle eigenvectors in U_j.

3: **end for**

4: Obtain the last core:

$$\hat{G}_d = \frac{1}{N} \sum_{i=1}^{N} D_d^{(i)} \in \mathbb{R}^{r_{d-1} \times n}$$

where $D_d^{(i)}$ is the reshaping of i-th column of D_d.

OUTPUT: Tensor train $\tilde{c} = \hat{G}_1 \circ \hat{G}_2 \circ \cdots \circ \hat{G}_d \in \mathbb{R}^{n \times n \times \cdots \times n}$.

3.4 TT-SVD with Kernel Nyström Approximation (TT-SVD-kn)

In this section, we propose a further modification based on the fast implementation of TT-SVD developed in Section 3.3.1. This modification accelerates Algorithm 4 using kernel

Nyström approximation to reduce the complexity from $\mathrm{O}(dN^2)$ to $\mathrm{O}(dN)$. As we have pointed out at the end of Section 3.3.1, the $\mathrm{O}(N^2)$ complexity of Algorithm 4 is the consequence of exact evaluation of the $N \times N$ matrix E_j in the step of eigen-decomposition for $B_j B_j^T$. The proposed modification in this section will utilize kernel Nyström approximation ([234]) to form E_j efficiently.

To begin with, we briefly introduce the idea of kernel Nyström method to approximate a symmetric matrix as shown in Figure 3.8. Given a symmetric matrix $A \in \mathbb{R}^{m \times m}$, the kernel Nyström method chooses $\tilde{r}$ columns of A uniformly at random without replacement and approximates A by the following low-rank cross approximation

$$A \approx M W^{+} M^T \tag{3.33}$$

where $M \in \mathbb{R}^{m \times \tilde{r}}$ consists of the $\tilde{r}$ chosen columns, $W \in \mathbb{R}^{\tilde{r} \times \tilde{r}}$ is the intersection of the chosen $\tilde{r}$ columns and the corresponding $\tilde{r}$ rows and W^{+} is the Moore-Penrose generalized inverse of W.

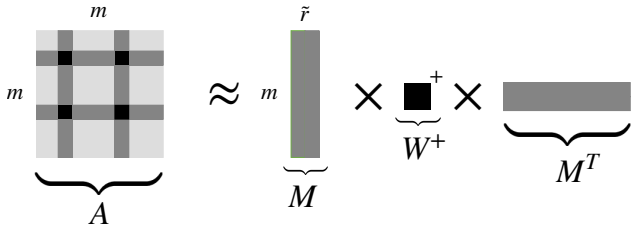


Figure 3.8: Cross approximation of a symmetric matrix.

Such approximation can be applied to the symmetric matrix E_j in Algorithm 4, resulting in the more efficient TT-SVD-kn algorithm summarized in Algorithm 3.4. By the recurrence relation (3.32), E_j is formulated as

$$E_j = (\tilde{\Phi}_d \tilde{\Phi}_d^T) \odot (\tilde{\Phi}_{d-1} \tilde{\Phi}_{d-1}^T) \odot \cdots \odot (\tilde{\Phi}_{j+1} \tilde{\Phi}_{j+1}^T), \tag{3.34}$$

which is depicted by the first equation in Figure 3.9. To apply kernel Nyström method, we

randomly sample $\tilde{r}$ rows in E_j with index $\mathcal{I} \subset \{1, 2, \cdots, N\}$, and we define $\tilde{\Phi}_m^{(\mathcal{I})} \in \mathbb{R}^{\tilde{r} \times n}$, which is denoted by the blocks in each $\tilde{\Phi}_m$ in Figure 3.9, as the resulting submatrices formed by the sampled rows.

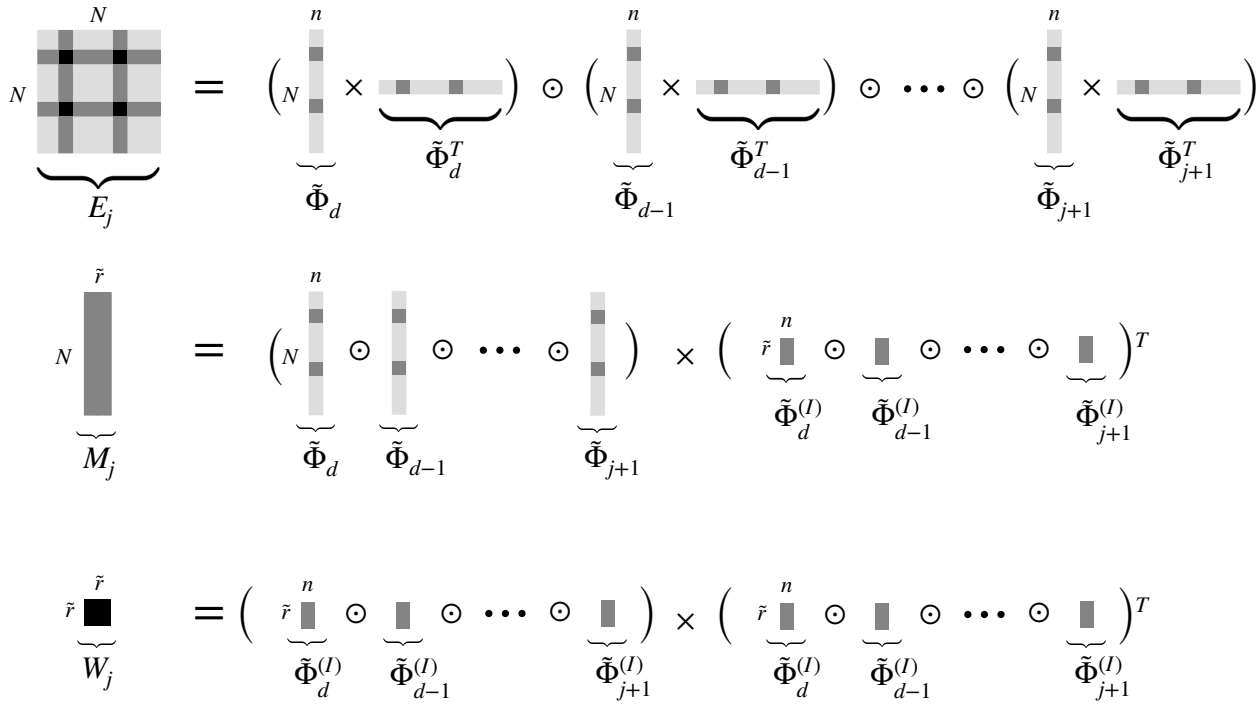


Figure 3.9: Cross approximation of E_j, the most expensive component in Algorithm 4.

At this point, we perform the cross approximation for E_j:

$$E_j \approx M_j W_j^+ M_j^T$$

where $M_j \in \mathbb{R}^{N \times \tilde{r}}$ is the submatrix formed by the chosen columns in E_j and $W_j \in \mathbb{R}^{\tilde{r} \times \tilde{r}}$ is the corresponding intersection. Using the expression (3.34), M_j and W_j are respectively formulated as the second and third equation in Figure 3.9. The structures of M_j and W_j allow us to compute them in a recursive manner as in Line 2 of Algorithm 5: suppose we have already obtained $\tilde{\Phi}_{j+2:d}$ and $\tilde{\Phi}_{j+2:d}^{(\mathcal{I})}$, M_j and W_j can be evaluated with complexity $\mathrm{O}(Nn)$ (direct computations based on their definitions in Figure 3.9 is $\mathrm{O}(dNn)$). Consequently,

$B_j B_j^T$ in the previous Algorithm 4 can now be approximated by

$$B_j B_j^T = \frac{1}{N^2} D_j E_j D_j^T \approx \frac{1}{N^2} D_j M_j W_j^+ M_j^T D_j^T$$

as in Line 4 of Algorithm 3.4. The computational cost for the above matrix multiplications is $\mathrm{O}(Nnr \max(nr, \tilde{r}))$, which is significantly smaller compared to the original $\mathrm{O}(N^2 nr)$ complexity in Algorithm 4. By considering the calculations for all d tensor cores, the total computational cost is $\mathrm{O}(dNnr \max(nr, \tilde{r}))$. Here we remark that we have employed the same indices $\mathcal{I}$ for computing all E_j. Such practice is the key to reuse the calculations in this recursive implementation so that we can achieve a complexity that is only linear in sample size.

Algorithm 5 TT-SVD-kn

INPUT: Samples $\{x^{(i)}\}_{i=1}^N$. Target rank $\{r_1, \cdots, r_{d-1}\}$. Number of sampled columns $\tilde{r}$.

1: Sample $\tilde{r}$ integers from $\{1, \cdots, N\}$ and let $\mathcal{I}$ be the set of sampled integers.

2: Pre-compute $\{W_j\}_{j=1}^{d-1}$ and $\{M_j\}_{j=1}^{d-1}$ based on sampled indices $\mathcal{I}$ recursively by Figure 3.9.

3: **for** $j \in \{1, \cdots, d-1\}$ **do**

4: Perform eigen-decomposition to $B_j B_j^T \in \mathbb{R}^{nr_{j-1} \times nr_{j-1}}$, which is approximated by

$$B_j B_j^T \approx A_j := \frac{1}{N^2} D_j M_j W_j^+ M_j^T D_j^T$$

where D_j is computed recursively as Figure 3.7. With eigen-decomposition $A_j = U_j \Lambda_j U_j^T$, we obtain the j-th core $\hat{G}_j \in \mathbb{R}^{r_{j-1} \times n \times r_j}$, which is reshaped from the first r_j principle eigenvectors in U_j.

5: **end for**

6: Obtain the last core:

$$\hat{G}_d = \frac{1}{N} \sum_{i=1}^{N} D_d^{(i)} \in \mathbb{R}^{r_{d-1} \times n}$$

where $D_d^{(i)}$ is the i-th column of D_d.

OUTPUT: Tensor train $\tilde{c} = \hat{G}_1 \circ \hat{G}_2 \circ \cdots \circ \hat{G}_d \in \mathbb{R}^{n \times n \times \cdots \times n}$.

3.5 TT-SVD with Truncated Cluster Basis (TT-SVD-c)

In this section, we introduce the TT-SVD-c algorithm (Algorithm 6), which is a modified TT-SVD algorithm based on truncated cluster basis sketching. The central idea is to utilize sketching technique to approximate the full SVDs of $B_j \in \mathbb{R}^{nr_{j-1} \times n^{d-j}}$ in Line 2 of Algorithm 3. More specifically, each step when we compute the left singular vectors U_j from B_j

in Algorithm 3, we instead compute the left singular vectors from a sketched B_j:

$$B'_j = B_j S_j, \tag{3.35}$$

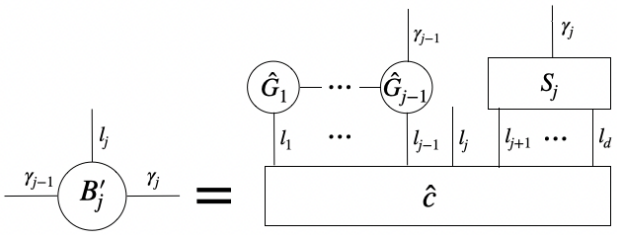
where S_j is some appropriately chosen sketching matrix with size $n^{d-j} \times \tilde{r}_j$, with sketching size $\tilde{r}_j$ much smaller than n^{d-j} (Line 2 of Algorithm 6). In this way, we can reduce the exponential complexity of full SVD for B_j to $\mathrm{O}(\max(nr\tilde{r}_j^2, n^2r^2\tilde{r}_j))$ complexity of SVD for B'_j.

Algorithm 6 TT-SVD-c

INPUT: Samples $\{x^{(i)}\}_{i=1}^N$. Target rank $\{r_1, \cdots, r_{d-1}\}$. Cluster order K.

1: **for** $j \in \{1, \cdots, d-1\}$ **do**

2: Perform SVD to $B'_j \in \mathbb{R}^{nr_{j-1} \times \tilde{r}_j}$, which is formed by the following tensor contraction



where $S_j \in \mathbb{R}^{n^{d-j} \times \tilde{r}_j}$ is the sketching matrix depending on cluster order K. The contraction of the righ-hand side is done efficiently as shown in Figure 3.10. With SVD $B'_j = U_j \Sigma_j V_j^T$, we obtain the j-th core $\hat{G}_j \in \mathbb{R}^{r_{j-1} \times n \times r_j}$, which is reshaped from the first r_j principle left singular vectors in U_j .

3: **end for**

4: Obtain the last core $\hat{G}_d = B'_d \in \mathbb{R}^{r_{d-1} \times n}$.

OUTPUT: Tensor train $\tilde{c} = \hat{G}_1 \circ \hat{G}_2 \circ \cdots \circ \hat{G}_d \in \mathbb{R}^{n \times n \times \cdots \times n}$.

The key of this approach is to choose good sketching matrix S_j such that B'_j retains the same range, and thus the same left singular space, as B_j. Here, we follow a specific way to

construct sketching matrix S_j ([37, 175]), where we only include the low-order cluster basis. We first introduce the set of *k-cluster indices*:

$$\mathcal{J}_{k,d} = \big\{(l_1, \cdots, l_d) \mid l_1, \cdots, l_d \in \{1, \cdots, n\} \quad \text{and} \quad \|l - \mathbf{1}\|_0 = k\big\},$$

which are the basis indices of those cluster bases in $\mathcal{B}_{k,d}$ defined in (3.5) since $\phi_1^j \equiv 1$.

Our choice of sketching matrix is based on the low-order cluster indices. To sketch B_j, we construct $S_j \in \mathbb{R}^{n^{d-j} \times \tilde{r}_j}$ as follows: each column of S_j is defined based on a multivariate index $l \in \bigcup_{k=0}^{K} \mathcal{J}_{k,d-j}$ where K is the chosen cluster order:

$$\mathbf{1}_{l_1} \otimes \mathbf{1}_{l_2} \otimes \cdots \otimes \mathbf{1}_{l_{d-j}}$$

where $\mathbf{1}_l$ is a $n \times 1$ column vector evaluating as 1 on the lth component and 0 elsewhere. Under this construction, S_j is a sparse matrix with column size

$$\tilde{r}_j = \sum_{k=0}^{K} |\mathcal{J}_{k,d-j}| = \mathrm{O}(d^K n^K).$$

At this point, the only concern is that forming $B'_j = B_j S_j$ has exponential complexity by direct matrix multiplication. Similar to Algorithm 4, we utilize the structure of $\hat{c}$ (3.16):

$$B'_j = B_j S_j = \frac{1}{N} D_j (E_j)^T = \frac{1}{N} \sum_{i=1}^{N} D_j^{(i)} (E_j^{(i)})^T \in \mathbb{R}^{n r_{j-1} \times \tilde{r}_j} \tag{3.36}$$

where D_j, E_j are defined in Figure 3.10 (c) and $D_j^{(i)}, E_j^{(i)}$ represent the i-th column of matrix D_j, E_j, respectively. According to the structure of D_j, E_j, their computation can be done in a recursive way following the diagram in Figure 3.10 (a) and (b).

In terms of the computational complexity of Algorithm 6, the cost for $\{E_j\}_{j=1}^{d-1}$ is $\mathrm{O}(d^{K+1} N n^K)$ for k-cluster truncation. Besides, the cost for $\{D_j\}_{j=1}^{d-1}$ follows the analy-

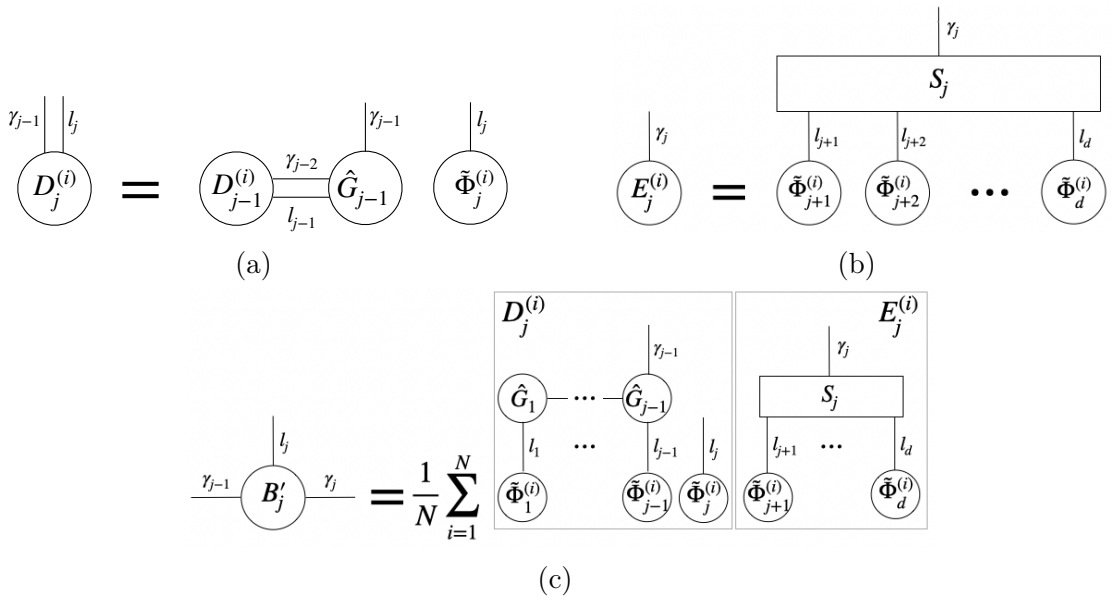


Figure 3.10: Tensor diagrams illustrating recursive computation of B'_j in Line 2 of Algorithm 6. (a) recursive calculation of $D_j^{(i)}$, which is the i-th column of the matrix D_j; (b) recursive calculation of $E_j^{(i)}$, which is the i-th column of the matrix E_j; (c) calculation of B'_j based on $D_j^{(i)}$ and $E_j^{(i)}$.

sis in Figure 3.7 and Algorithm 4, which is $\mathrm{O}(dNnr^2)$. Furthermore, the computational complexity of multiplication in B'_j and the following SVD step requires $\mathrm{O}(d^{K+1}Nn^{K+1})$. To summarize, the dominant term in computational complexity is $\mathrm{O}(d^{K+1}Nn^{K+1})$. The reduction is from standard TT-SVD algorithm $\mathrm{O}(\min(dn^{d+1}, dN^2n))$ to $\mathrm{O}(d^{K+1}Nn^{K+1})$, which is significant especially for a large sample size N and low-order K. In the following subsection, we further propose a modification of TT-SVD-c which constructs sketching matrix S_j based on hierarchical matrix decomposition. This acceleration can achieve a lower computational complexity as $\mathrm{O}(d\log(d))$.

3.5.1 Acceleration of TT-SVD-c via Hierarchical Matrix Decomposition

In this subsection, we propose a modification that further accelerates TT-SVD-c method with cluster order $K = 1$. The key insight of this acceleration lies in the assumption that

the ground truth density p^* has high correlations between nearby variables (such as x_1 and x_2) and low correlation between distant variables (such as x_1 and x_d), which is satisfied by many real cases. In Algorithm 6, the sketching matrix S_j considers all correlations equally without taking the distance into account, which includes redundant correlations with distant variables and leads to a large column size of S_j. In fact, even if cluster order $K = 1$, the column size is $\mathrm{O}(d)$, which results in the overall computational cost of $\mathrm{O}(d^2)$. This motivates us to further sketch S_j, which is done by hierarchical matrix decomposition ([79, 80, 16]) of covariance matrix M, whose (j_1, j_2)-block is formulated as

$$M_{j_1,j_2} = \left[\mathbb{E}_{\hat{p}}[\phi_{l_1}^{j_1}(x_{j_1})\phi_{l_2}^{j_2}(x_{j_2})]\right]_{l_1,l_2} \in \mathbb{R}^{n\times n} \tag{3.37}$$

for all $1 \leqslant j_1, j_2 \leqslant d$.

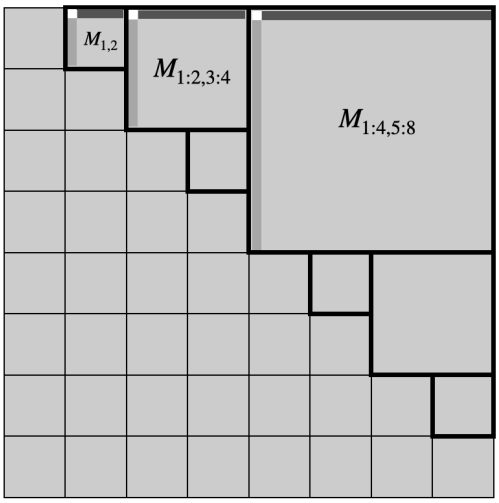


Figure 3.11: Hierarchical matrix decomposition of covariance matrix for $d = 8$. The figure depicts the three block matrices $M_{1:4,5:8}, M_{1:2,3:4}, M_{1,2}$ whose top $\tilde{r}$ right singular vectors (visualized as dark gray matrix in each block matrix).

To illustrate the idea, we consider an example with $d = 8$. Covariance matrix M is depicted in Figure 3.11, where we use $M_{j_1:j_2,j_3:j_4}$ to represent the submatrix formed by blocks with row index from j_1 to j_2 and column index from j_3 to j_4. To implement hierarchical matrix decomposition, we perform SVD on upper triangular off-diagonal blocks

$M_{1:4,5:8}, M_{1:2,3:4}, M_{1,2}$ and get top $\tilde{r}$ right singular vectors V_1', V_2', V_3', respectively. Here the sketching size $\tilde{r}$ (different from that in Algorithm 6) is chosen as a constant. V_1', V_2', V_3' respectively capture correlations with variables $x_{5:8}$, $x_{3:4}$ and x_2.

Once the singular vectors V_1', V_2', V_3' are obtained, we can use them to construct sketching matrices with smaller column sizes. When computing the first core $\hat{G}_1$, we form $Q_1 = \text{diag}[V_1', V_2', V_3'] \in \mathbb{R}^{7n \times 3\tilde{r}}$ which captures the weighted correlation between x_1 and $x_{2:8}$, and we use S_1 and Q_1 to sketch B_1 as $B_1' = B_1 S_1 Q_1 \in \mathbb{R}^{n \times 3\tilde{r}}$. Once this is done, we perform SVD to B_1' to obtain $\hat{G}_1$. For the second core, we form $Q_2 = \text{diag}[V_1', V_2'] \in \mathbb{R}^{6n \times 2\tilde{r}}$ to capture the weighted correlation between variables x_2 and variables $x_{3:8}$. Then we sketch $B_2' = B_2 S_2 Q_2$, and perform SVD to B_2' to obtain $\hat{G}_2$. Note that we can reuse V_1', V_2' in the second step since both of them were already computed in the first step. By repeating the procedure and generating new sketch $B_j' = B_j S_j Q_j$ for the remaining j, this hierarchical approach enables the efficient computation of all cores $\{\hat{G}_j\}_{j=1}^d$.

In terms of computational complexity, there are two major sources of computations. The first is the SVD of off-diagonal block matrices ($M_{1:4,5:8}, M_{1:2,3:4}, M_{1,2}$ for $d = 8$). For each block, we first apply kernel Nyström approximation (3.33) with $\tilde{r}$ sampled rows and columns, followed by SVD. The cost for this part of computations is $\text{O}(d \log(d) N n \tilde{r})$. The second major computation is forming $B_j' = B_j S_j Q_j$. The number of columns in the weight matrix Q_j is at most $\log_2(d)\tilde{r}$. Since S_j is a sparse matrix, the computation of $B_j' = B_j S_j Q_j$ for all j needs $\text{O}(d \log(d) N n \tilde{r})$ computational cost upon leveraging a similar strategy for computing (3.36). The computational costs discussed above are the dominant cost in Algorithm 6. We can replace S_j with $S_j Q_j$ in Algorithm 6 and follow the operations in Figure 3.10, the method achieves a reduced computational complexity $\text{O}(d \log(d) N n)$ compared to the original complexity $\text{O}(d^2 N n^2)$ for TT-SVD-c with cluster order $K = 1$.

3.6 TT-SVD with Random Tensorized Sketching (TT-rSVD-t)

In this section, we propose another modification for TT-SVD algorithm via sketching. This approach follows the randomized sketching technique ([100]), originated from randomized linear algebra literature ([82]), to accelerate SVD. The key idea is to choose sketching matrix $S_j \in \mathbb{R}^{n^{d-j} \times \tilde{r}}$ as some random matrix, where $\tilde{r}$ is a given sketching size, which is usually chosen to be reasonably larger than target rank r of the output tensor-train estimator. Afterwards, similar as TT-SVD-c, we perform SVD to $B'_j = B_j S_j$. In general, the matrix multiplication $B_j S_j$ has exponentially large computational cost. Our solution to this curse of dimensionality is to form S_j by a set of random tensor trains, leading to the TT-rSVD-t algorithm summarized in Algorithm 7.

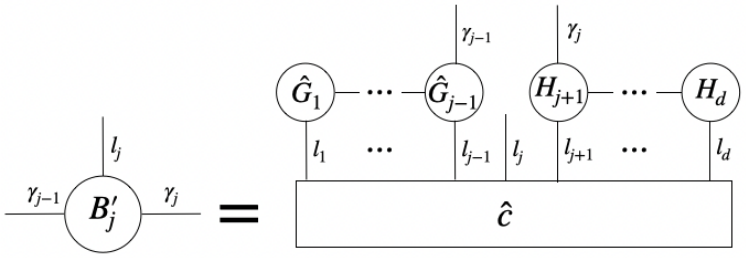
Algorithm 7 TT-rSVD-t

INPUT: Samples $\{x^{(i)}\}_{i=1}^N$. Target rank $\{r_1, \cdots, r_{d-1}\}$. Sketching size $\tilde{r}$.

1: Generate random tensor train with cores $\{H_j\}_{j=1}^d$, where $H_1 \in \mathbb{R}^{n\times\tilde{r}}, H_2, \cdots, H_{d-1} \in \mathbb{R}^{\tilde{r}\times n\times\tilde{r}}, H_d \in \mathbb{R}^{\tilde{r}\times n}$.

2: **for** $j \in \{1, \cdots, d-1\}$ **do**

3: Perform SVD to $B'_j \in \mathbb{R}^{nr_{j-1}\times\tilde{r}_j}$, which is formed by the following tensor contraction



The contraction of the righ-hand side is done efficiently as shown in Figure 3.12. With SVD $B'_j = U_j\Sigma_j V_j^T$, we obtain the j-th core $\hat{G}_j \in \mathbb{R}^{r_{j-1}\times n\times r_j}$, which is reshaped from the first r_j principle left singular vectors in U_j .

4: **end for**

5: Obtain the last core $\hat{G}_d = B'_d \in \mathbb{R}^{r_{d-1}\times n}$.

OUTPUT: Tensor train $\tilde{c} = \hat{G}_1 \circ \hat{G}_2 \circ \cdots \circ \hat{G}_d \in \mathbb{R}^{n\times n\times\cdots\times n}$.

To implement Algorithm 7, we first generate a d-dimensional random tensor train with fixed rank $\tilde{r}$ whose cores are $H_1, H_2, \cdots, H_d$ (Line 1). In practice, we can set each core as i.i.d. random Gaussian matrix/tensor or random uniform matrix/tensor. When sketching B_j, we construct sketching matrix $S_j \in \mathbb{R}^{n^{d-j}\times\tilde{r}}$ as the reshaping of $H_{j+1} \circ H_{j+2} \circ \cdots \circ H_d$ (Line 3). Afterwards, we contract B_j with S_j and perform SVD to solve the j-th core $\hat{G}_j$.

Similar to (3.36) for TT-SVD-c, B'_j is computed recursively as illustrated in Figure 3.12(c). The key difference lies in the new formulation of E_j, which arises from the new choice of sketching matrix S_j. Leveraging the tensor-train structure, E_j can also be computed recursively as shown in Figure 3.12(b). The total computational cost for $\{E_j\}_{j=1}^{d-1}$ is $\mathrm{O}(dNn\tilde{r}^2)$.

The computation of $\{D_j\}_{j=1}^{d-1}$, depicted in Figure 3.12(a), follows the same approach used in Algorithm 4 and incurs a computational cost of $\mathrm{O}(dNnr^2)$. Overall, the SVD step required to compute all tensor cores has a total complexity of $\mathrm{O}(dNn\tilde{r}^2)$, which is linear with respect to both dimensionality and sample size.

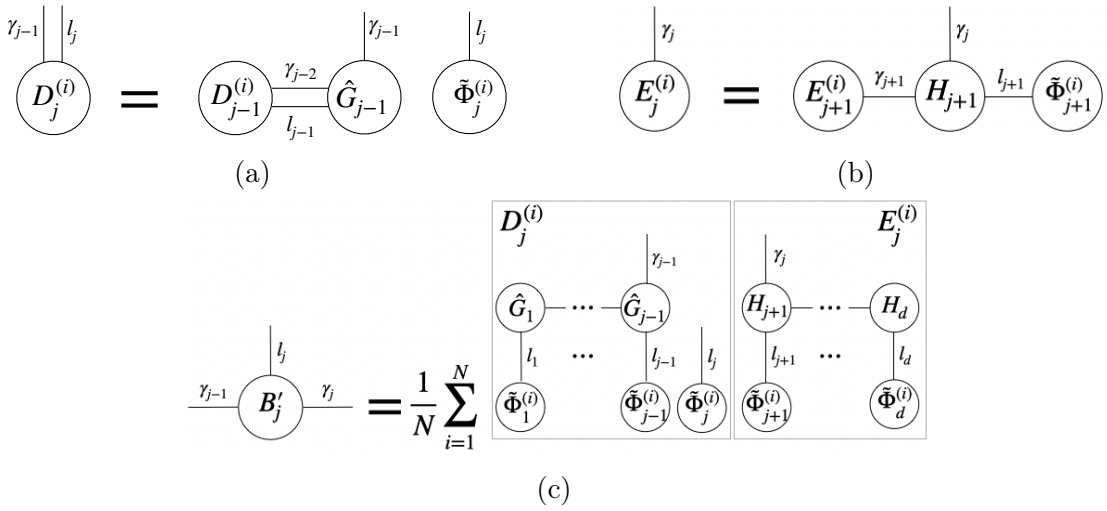


Figure 3.12: Tensor diagrams illustrating recursive computation of B'_j in Line 2 of Algorithm 7. (a) recursive calculation of $D_j^{(i)}$, which is the i-th column of the matrix D_j; (b) recursive calculation of $E_j^{(i)}$, which is the i-th column of the matrix E_j; (c) calculation of B'_j based on $D_j^{(i)}$ and $E_j^{(i)}$.

3.7 Pre-processing Steps for General Distributions

In this section, we propose several pre-processing steps to extend the discussed framework to general distributions. The pre-processing procedure consists of a dimension-reduction step using principal component analysis (PCA), followed by a step to determine the mean-field and the univariate bases. The full algorithm is summarized in Algorithm 8.

Algorithm 8 Tensor density estimator after variable re-ordering.

INPUT: Samples $\{x^{(i)}\}_{i=1}^N$. Embedding dimension d', $d' \leqslant d$. Target rank $\{r_1, \cdots, r_{d-1}\}$.

1: Apply PCA to the samples $\{x^{(i)}\}_{i=1}^N$. Denote the top d' orthonormal principal components as $Q \in \mathbb{R}^{d \times d'}$. Define the new variables as $z^{(i)} = Q^T x^{(i)}$, $i = 1, \cdots, N$, with the associated empirical distribution denoted by $\hat{p}_z$.

2: Estimate the 1-marginals $\mu_1'(z_1) \cdots \mu_{d'}'(z_{d'})$ using one-dimensional kernel density estimation. The product of these 1-marginals forms the mean-field $\mu' = \prod_{j=1}^{d'} \mu_j'$ in the reduced coordinate system.

3: Compute the orthogonal basis $\{\phi_l^j(z_j)\}_{l=1}^n$ for each $j = 1, \cdots, d'$, where the basis functions are orthogonalized with respect to $\mu_j'(z_j)$.

4: Obtain the d'-dimensional tensor density estimator in the reduced coordinate system:

$$\tilde{p}_z = \mu' \odot \operatorname{diag}(\tilde{w}_\alpha)^{-1} \Phi \left(\texttt{TT-compress} \left(\operatorname{diag}(\tilde{w}_\alpha) \Phi^T \hat{p}_z \right) \right),$$

where `TT-compress` is any tensor-train compression algorithm detailed in Section 3.3 – 3.6. The final density estimator in the original space is obtained as:

$$\tilde{p}(x) = \frac{1}{Z} \tilde{p}_z(Q^T x),$$

where Z is the normalization factor.

OUTPUT: Tensor density estimator $\tilde{p}$.

When dealing with a general distribution, the optimal ordering of variables is generally unknown. A poor variable ordering may lead to a high-rank TT representation, significantly increasing computational complexity. To address this, Step 1 of Algorithm 8 applies PCA to simultaneously decorrelate the variables and reduce the dimensionality. The resulting reduced-coordinate variables $\{z^{(i)}\}_{i=1}^N$ often exhibit simpler dependence structures, allowing a more compact TT representation. The eigen-decomposition in PCA can be performed

through a "streaming approach", as in [89, 98, 111] with a complexity of $O(Ndd')$.

In Step 2 of Algorithm 8, we aim to perform the projection (3.10) on the empirical distribution $\hat{p}_z$. However, to do so, we require knowledge of the mean-field $\mu'(z) = \prod_{j=1}^{d'} \mu'_j(z_j)$. To approximate this, we apply a kernel density estimator to each of the one-dimensional marginals of $\hat{p}_z$. This step ensures that μ' captures the product structure of the new coordinate system.

In Step 3, we construct orthonormal basis functions with respect to each one-dimensional marginal. This can be achieved using a Gram-Schmidt procedure or a stable three-term recurrence method (see [65]) for polynomial basis functions.

Finally, in Step 4, we estimate the density $\hat{p}_z$ in the reduced-coordinate system using the tensor-train density estimator. This involves applying any tensor-train compression algorithms `TT-compress` detailed in Section 3.3 – 3.6 to $\mathrm{diag}(\tilde{w}_\alpha)\Phi^T\hat{p}_z$. The resulting density estimator $\tilde{p}_z$ is then mapped back to the original space via the transformation $\tilde{p}(x) = \tilde{p}_z(Q^T x)$ followed by normalization.

3.8 Theoretical Results

In this section, we present an error analysis of our proposed methods. For simplicity, we focus primarily on the hard-thresholding approach described in (3.9). Based on Section 3.2.3, we know the estimator from hard-thresholding approach in (3.9) is close to the primarily discussed soft-thresholding approach in (3.10). The detailed analysis of the soft-thresholding approach in (3.10) will be deferred to future work.

For simplicity, we assume the mean-field component $\mu \sim \mathrm{Unif}([0,1]^d)$. In this way, the univariate basis functions $\{\phi_l\}_{l=1}^n$ are orthonormal, and we take Legendre polynomials as an example in our analysis. In addition, we propose the following assumptions on the ground truth density:

Assumption 4. *Suppose the ground truth density function $p^* : [0,1]^d \to \mathbb{R}^+$ satisfies is an additive model (1-cluster):*

$$p^*(x_1, \cdots, x_d) = 1 + q_1^*(x_1) + \cdots + q_d^*(x_d), \tag{3.38}$$

where $q_j^* \in H^{\beta_0}([0,1])$ for some integer $\beta_0 \geqslant 1$ satisfies the regularity

$$\min_j \|q_j^*\|_{L_2([0,1])} \geqslant b^* \quad \textit{and} \quad \max_j \|q_j^*\|_{H^{\beta_0}([0,1])} \leqslant B^*$$

for two $\mathrm{O}(1)$ positive constants b^* and B^* where we set $B^* \geqslant 1$. Under this assumption, we can immediately obtained that $\|p^*\|_\infty \leqslant 1 + B^* d$ according to Sobolev embedding theorem.

Assumption 4 considers only the 1-cluster model and include the regularity constraint for each one-dimensional component. This motivation stems from the fact that the left/right subspaces are computationally straightforward, allowing for an efficient tensor-train representation of p^*. This specific structure, in turn, yields a favorable estimation error rate with respect to the dimensionality.

We state our main theorem under these assumptions:

Theorem 4. *Let $c^* = diag(w_K)\Phi^T p^*$ and $\hat{c} = diag(w_K)\Phi^T \hat{p}$ be the $\mathbb{R}^{n^d}$ coefficient tensor associated with the ground truth and empirical density function, respectively. Suppose the ground truth density function p^* satisfies Assumption 4, and the sample size is sufficiently large such that*

$$N \geqslant C_0 d^{2K-1} n^{4K-2} \log(dn), \tag{3.39}$$

where C_0 is some sufficiently large constant independent of N, n and d. When we use TT-SVD (Algorithm 3) as `TT-compress` *to compress $\hat{c}$ to $\tilde{c}$ with target TT rank $r = 2$, the*

following bound holds with high probability:

$$\frac{\|\tilde{c} - c^*\|_F}{\|c^*\|_F} = \mathrm{O}\left(\sqrt{\frac{dnB^* \log(dn)}{N} \sum_{j=1}^{d-1} \frac{1}{\sigma_2^2(c^{*(j)})}}\right) \tag{3.40}$$

where $c^{*(j)}$ is the jth-unfolding matrix of c^* as defined in (3.11)*, and $\sigma_2(\cdot)$ denotes the second largest singular value.*

The proof of this theorem is lengthy and therefore deferred to Section B.2.1. The theorem states that the relative error of the coefficient estimates decreases at the Monte Carlo rate of $\mathrm{O}(1/\sqrt{N})$ with respect to the sample size N. Furthermore, under the assumption that p^* is a 1-cluster, the relative error grows only linearly with the dimensionality d (up to a logarithmic factor), provided that the second-largest singular values (also the smallest since $r = 2$) of all reshaped ground truth tensors c^* are uniformly bounded below. As a direct consequence of this theorem, one can derive the final error estimate for the density estimator:

Corollary 1. *Suppose all the conditions in Theorem 4 hold, the following bound for density estimator $\tilde{p}$* (3.24) *by TT-SVD (Algorithm 3) with target rank $r = 2$ holds with high probability:*

$$\frac{\|\tilde{p} - p^*\|_{L_2([0,1]^d)}}{\|p^*\|_{L_2([0,1]^d)}} = \mathrm{O}\left(\frac{B^*}{b^*}\Big(n^{-\beta_0} + \frac{B^*}{b^{*2}}\sqrt{\frac{dn\log^2(dn)}{N}}\Big)\right), \tag{3.41}$$

where β_0 is the regularity parameter specified in Assumption 4.

The proof of Corollary 1 can be found in Section B.2.2.

3.9 Numerical Results

In this section, we carry out numerical experiments to examine the performance of our proposed density estimation algorithms. In specific, we use TT-SVD-kn (Algorithm 5), TT-SVD-c (Algorithm 6) with hierarchical matrix decomposition and TT-rSVD-t (Algorithm

7). Unless otherwise noted, we use the following parameter settings throughout: we set the number of sampled columns in Nyström approximation $\tilde{r} = 100$ in TT-SVD-kn, sketching size $\tilde{r} = 10$ in TT-SVD-c with hierarchical matrix decomposition, sketching size $\tilde{r} = 30$ in TT-rSVD-t and the kernel parameter $\alpha = 0.01$. These parameter choices are based on empirical observations.

For simplicity, we choose the mean-field density μ as a uniform distribution, so that $\{\phi_l^j\}_{l=1}^n$ can be chosen as any orthonormal basis functions. Here we use the Fourier basis

$$\phi_l(x) = \begin{cases} \frac{1}{\sqrt{2L}}, & \text{for } l = 1, \\ \frac{1}{\sqrt{L}} \cos(\frac{l\pi x}{2L}) & \text{for even } l > 1 \\ \frac{1}{\sqrt{L}} \sin(\frac{(l+1)\pi x}{2L}) & \text{for odd } l > 1 \end{cases} \tag{3.42}$$

as the univariate basis function for all $j = 1, \cdots, d$.

3.9.1 Gaussian Mixture Model

We first consider the d-dimensional Gaussian mixture model which is defined as :

$$\frac{1}{6}\mathcal{M}(0.18) + \frac{1}{3}\mathcal{M}(0.2) + \frac{1}{2}\mathcal{M}(0.22)$$

where

$$\mathcal{M}(\sigma) = \frac{2}{3}\mathcal{N}(-0.5, \sigma^2 I_{d\times d}) + \frac{1}{3}\mathcal{N}(0.5, \sigma^2 I_{d\times d}).$$

We set the domain as the hypercube $[-L, L]^d$ where $L = 1.5$ with mesh size 0.1 in each direction.

We first fix dimension $d = 10$. We draw N i.i.d samples from the mixture model, and use proposed three algorithms to fit a TT approximation for the empirical distribution of these samples. For the algorithms, we set fixed target rank $r = 3$ and use $n = 17$ Fourier bases. In

the left panel of Figure 3.13, we test different sample sizes N and plot the relative L_2-error defined by

$$\|\tilde{p} - p^*\|_{L_2} / \|p^*\|_{L_2}$$

in logarithm scale. One can observe that the relative error decays with Monte Carlo rate $\mathrm{O}(1/\sqrt{N})$. Here we remark that the ground truth p^* for this Gaussian mixture model can be formulated as a TT without approximation error, and the numerical L_2-norm based on quadrature of TTs can be easily computed with $\mathrm{O}(d)$ complexity [166].

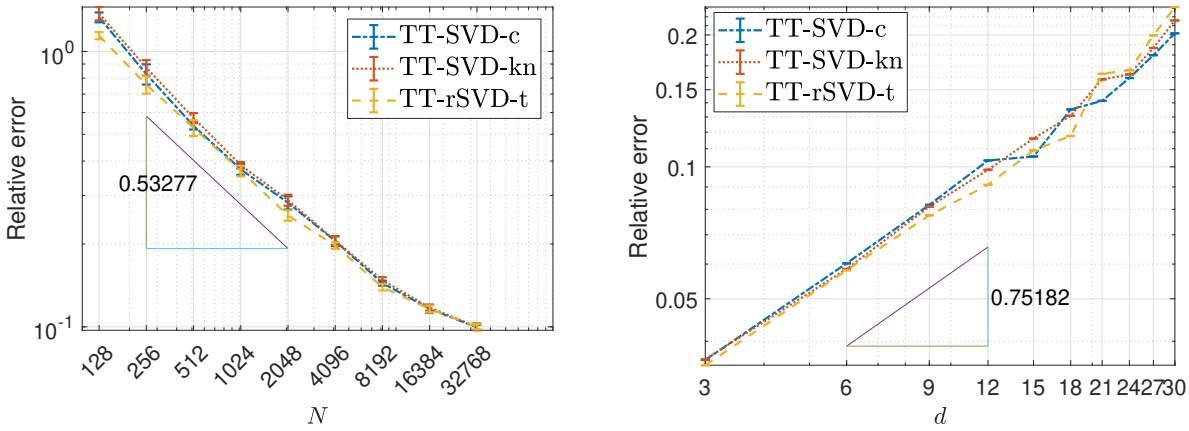


Figure 3.13: Gaussian mixture model: relative L_2-error for proposed algorithms. Left: $d = 10$ and sample number $N = 2^7, 2^8, \cdots, 2^{15}$. Right: $N = 10^6$ and dimension $d = 3, 6, 9, \cdots, 30$.

Next, we fix the sample size $N = 10^6$ and test our algorithms with various dimensions d. The relative L_2-error is plotted in right panel of Figure 3.13, which exhibits sublinear growth with increasing dimensionality according to the curves. In particular, with $N = 10^6$ samples, our algorithms can achieve two-digit relative L_2-error up to $d = 12$. For dimensionality higher than that, we can expect that the error can also be controlled below 0.1 for a sufficiently large sample size, as suggested by the Monte Carlo convergence observed in the previous experiment.

Finally, we verify the time complexity of the algorithms. In Figure 3.14, we plot the wall clock time of three algorithms for fixed $d = 5$ with various N (left panel) and fixed

$N = 10000$ with various d (right panel). The rates of wall clock time verify the theoretical computational complexity at listed in Table 3.1.

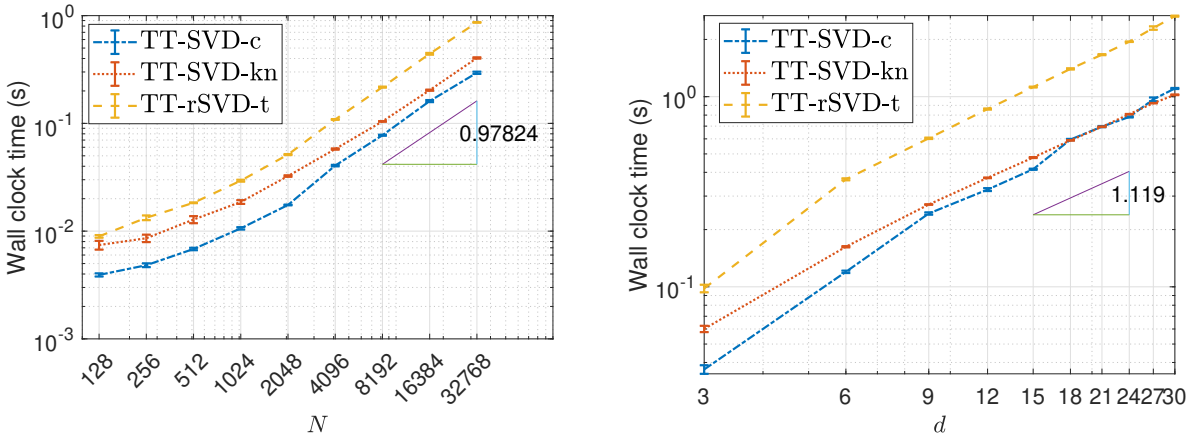


Figure 3.14: Gaussian mixture model: average wall clock time (seconds) for proposed algorithms. Left: $d = 5$ and sample number $N = 2^7, 2^8, \cdots, 2^{15}$. Right: $N = 10000$ and dimension $d = 3, 6, 9, \cdots, 30$.

3.9.2 Ginzburg-Landau Model

Now we consider a more practical problem in statistical physics as the Ginzburg-Landau model [94]. In particular, we consider the density estimation for the Boltzmann distribution

$$p^*(x) = \frac{1}{Z}\mathrm{e}^{-\beta V(x)}$$

where β is the inverse temperature, Z is the normalization factor such that $\int p^*(x)dx = 1$, and V is the Ginzburg-Landau potential energy.

1D Ginzburg-Landau model

We first consider a 1D Ginzburg-Landau model where the potential energy is formulated as

$$V(x_1, \cdots, x_d) = \sum_{i=1}^{d+1} \frac{\lambda}{2}\left(\frac{x_i - x_{i-1}}{h}\right)^2 + \frac{1}{4\lambda}(1 - x_i^2)^2 \text{ for } x \in [-L, L]^d \tag{3.43}$$

where $x_0 = x_{d+1} = 0$, $h = 1/(1+d)$, $\lambda = 0.03$ and $\beta = 1/8$. We consider hypercube domain $[-L, L]^d$ with $L = 2.5$ and mesh size 0.05 in each direction. For the ground truth p^*, we use DMRG-cross algorithm (see [190]) to get p^* in TT form, which is known to provide highly accurate approximations with arbitrarily prescribed precision. The samples of p^* are generated by running Langevin dynamics for sufficiently long time.

In Table 3.2, we list the relative L_2-error of the approximated densities estimated by $N = 10^6$ samples with different number of univariate basis functions. For this example, we set fixed target rank $r = 4$. One can observe that the approximation improves as we use more basis functions. Such improvement can be further visualized in Figure 3.15, where we plot the marginal distribution of the approximated solutions solved by three algorithms when $d = 20$. All of them give accurate match to the sample histograms when a sufficient number of basis functions are used.

n	$d = 5$			$d = 10$		
	TT-SVD-c	TT-SVD-kn	TT-rSVD-t	TT-SVD-c	TT-SVD-kn	TT-rSVD-t
7	0.2810	0.2809	0.2810	0.4304	0.4304	0.4305
11	0.0782	0.0781	0.0781	0.1365	0.1365	0.1378
15	0.0305	0.0304	0.0301	0.0488	0.0483	0.0533
19	0.0275	0.0273	0.0289	0.0449	0.0404	0.0526
n	$d = 15$			$d = 20$		
	TT-SVD-c	TT-SVD-kn	TT-rSVD-t	TT-SVD-c	TT-SVD-kn	TT-rSVD-t
7	0.6342	0.6342	0.6343	0.7897	0.7897	0.7898
11	0.2169	0.2167	0.2186	0.3017	0.3016	0.3037
15	0.0733	0.0740	0.0836	0.1126	0.1125	0.1158
19	0.0613	0.0602	0.0716	0.0884	0.0874	0.1099

Table 3.2: 1D G-L model: comparison of L_2 relative errors with $N = 10^6$ and $d = 5, 10, 15, 20$ under different numbers of univariate basis functions used.

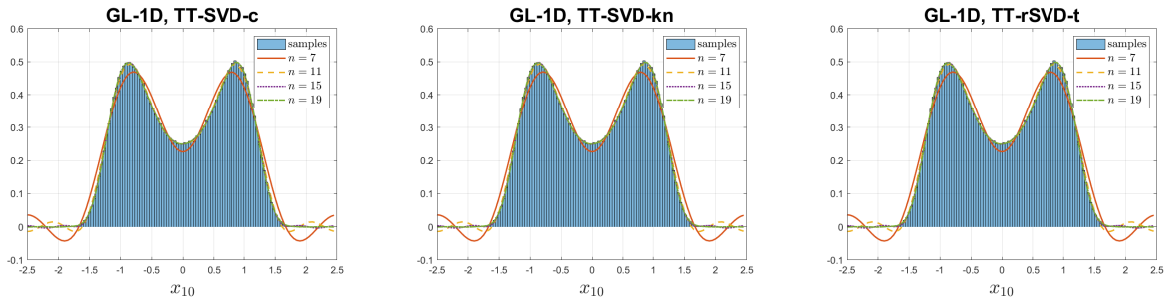


Figure 3.15: 1D G-L model: x_{10}-marginal of approximated solutions for $d = 20$ and $N = 10^6$ under different numbers of univariate basis functions used.

2D Ginzburg-Landau model

Next, we consider the 2D Ginzburg-Landau model which is defined on a $m \times m$ lattice. The potential energy is formulated as

$$V(x_{1,1}, \cdots, x_{m,m}) = \frac{\lambda}{2} \sum_{i=1}^{m+1} \sum_{j=1}^{m+1} \left[\left(\frac{x_{i,j} - x_{i-1,j}}{h} \right)^2 + \left(\frac{x_{i,j} - x_{i,j-1}}{h} \right)^2 \right] + \frac{1}{4\lambda} \sum_{i=1}^{m} \sum_{j=1}^{m} (1 - x_{i,j}^2)^2$$

with the boundary condition

$$x_{0,j} = x_{m+1,j} = 1, \quad x_{i,0} = x_{i,m+1} = -1 \quad \forall\, i, j = 1, \cdots, m$$

where we choose parameter $h = 1/(1+m)$, $\lambda = 0.1$ and inverse temperature $\beta = 1$. We set $L = 2$ with mesh size 0.05 for the domain. We consider a test example where $m = 16$ and thus the total number of dimensionality is $m^2 = 256$. For this 2D model, we apply the PCA pre-processing steps described in Section 3.7 to order the dimensions and also reduce the dimensionality, where we choose embedding dimension $d' = 100$. In terms of the algorithms, we set target rank $r = 10$, $\alpha = 0.001$ and we use $N = 10^4$ samples and $n = 21$ Fourier bases.

We evaluate the performance of our fitted tensor-train density estimators as generative models. Specifically, we first fit tensor-train density estimators $\tilde{p}$ from the given $N = 10^4$ samples $\{x_i\}_{i=1}^N$ using proposed algorithms, then generate new samples $\{\tilde{x}_i\}_{i=1}^N$ from the

learned tensor trains. Sampling each $\tilde{x}_i \sim \tilde{p}$ can be done via tensor-train conditional sampling

$$\tilde{p}(x_1, x_2, \cdots, x_d) = \tilde{p}(x_1)\tilde{p}(x_2|x_1)\tilde{p}(x_3|x_2, x_1)\cdots\tilde{p}(x_d|x_{d-1}, \cdots, x_1) \tag{3.44}$$

where each

$$\tilde{p}(x_i|x_{i-1}, \cdots, x_1) = \frac{\tilde{p}(x_1, \cdots, x_i)}{\tilde{p}(x_1, \cdots, x_{i-1})}$$

is a conditional distribution of $\tilde{p}$, and $\tilde{p}(x_1), \cdots, \tilde{p}(x_1, \cdots, x_{d-1})$ are the marginal distributions. We point out each conditional sampling can be done in $\mathrm{O}(d)$ complexity using the TT structure of $\tilde{p}$ [55].

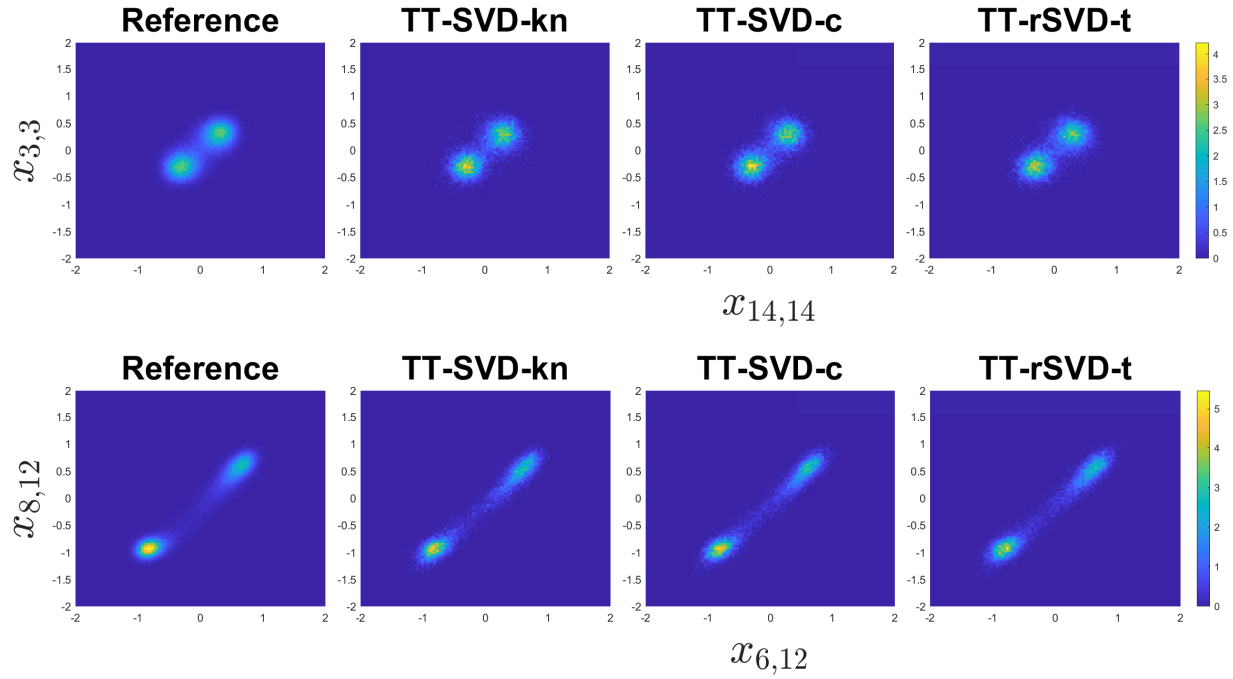


Figure 3.16: 2D G-L model on a 16×16 lattice: visualization of $(x_{3,3}, x_{14,14})$-marginal (first row) and $(x_{6,12}, x_{8,12})$-marginal (second row). Left to right: reference sample histograms, samples from tensor trains fitted by proposed algorithms.

Figure 3.16 illustrates the $(x_{3,3}, x_{14,14})$-marginal and $(x_{6,12}, x_{8,12})$-marginal distributions of the generated samples. For comparison, we also show results for $N^* = 10^5$ reference samples. The visualizations reveal that samples generated from tensor trains learned by all

three proposed algorithms can capture local minima accurately. To quantify the accuracy, we compute the second-moment error

$$\|\tilde{X}^T\tilde{X} - X^{*T}X^*\|_F / \|X^{*T}X^*\|_F \tag{3.45}$$

where $\tilde{X} \in \mathbb{R}^{N\times d}$ is the matrix formed by the generated samples, and $X^* \in \mathbb{R}^{N^*\times d}$ are reference samples. The second-moment errors for TT-SVD-kn, TT-SVD-c and TT-rSVD-t algorithms are respectively $0.0278, 0.0252, 0.0264$.

3.9.3 Comparison with Neural Network Approaches

In this section, we compare the performance of generative models fitted by our proposed tensor-train algorithms with one example of neural network-based generative methods ([170]). For neural network details, we use 5 transforms and each transform is a 3-hidden layer neural network with width 128. We use Adam optimizer to train the neural network. The experiments are based on selected examples from previous subsections as well as real data from MNIST.

1D Ginzburg-Landau model

We first consider the Boltzmann distribution of 1D Ginzburg-Landau potential defined in (3.9.2) where we choose model parameters $\lambda = 0.005$ and $\beta = 1/20$. We use $N = 10^4$ samples as training data to fit tensor trains, and then generate $N = 10^4$ new samples from the learned tensor trains by conditional sampling (3.44).

In Figure 3.17, we plot the second-moment errors of the samples generated by tensor-train algorithms and a neural network, compared against a reference set of $N^* = 10^6$ samples across various dimensions $d = 10, 20, \cdots, 100$. Here we choose target rank $r = 4$ and use $n = 17$ Fourier bases for all simulations. The three proposed tensor-train algorithms

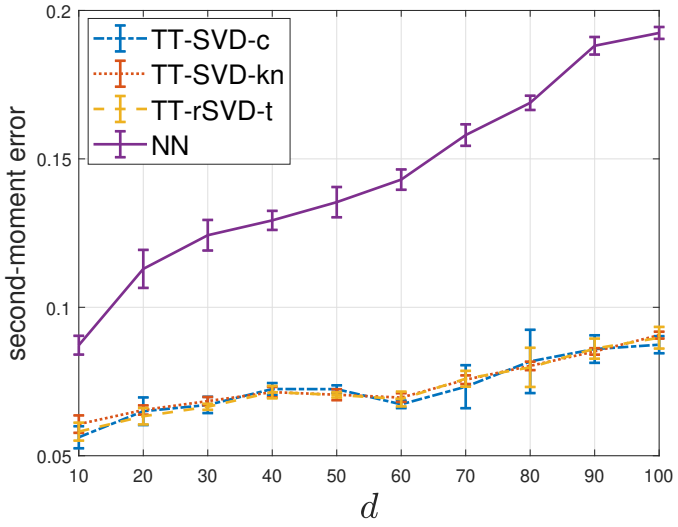


Figure 3.17: Comparison of second-moment errors of generated samples.

exhibit similar performance and consistently achieve two-digit accuracy across all dimensions. Moreover, they are considerably more accurate than the neural network. To visualize this accuracy gap, Figure 3.18 further shows histograms of two selected marginal distributions from the generated samples for $d = 100$. It can be observed that samples generated by the tensor-train estimators more accurately capture local minima compared to those produced by the neural network.

Real data: MNIST

We also evaluate our algorithms on the MNIST dataset, which consists of 70,000 images of handwritten digits (0 through 9), each labeled with its corresponding digit. Each image is represented as a 28×28 pixel matrix. To learn the generative models, we treat the dataset as a collection of 70000×28^2 samples. For each digit class, we apply the TT-rSVD-t algorithm with parameters $\alpha = 0.05$, sketching rank $\tilde{r} = 40$, target rank $r = 10$ and $n = 15$ Fourier bases, to learn tensor-train representations from the corresponding samples of dimension $d = 28^2 = 784$. Again, the PCA pre-processing steps are applied to reduce dimensionality,

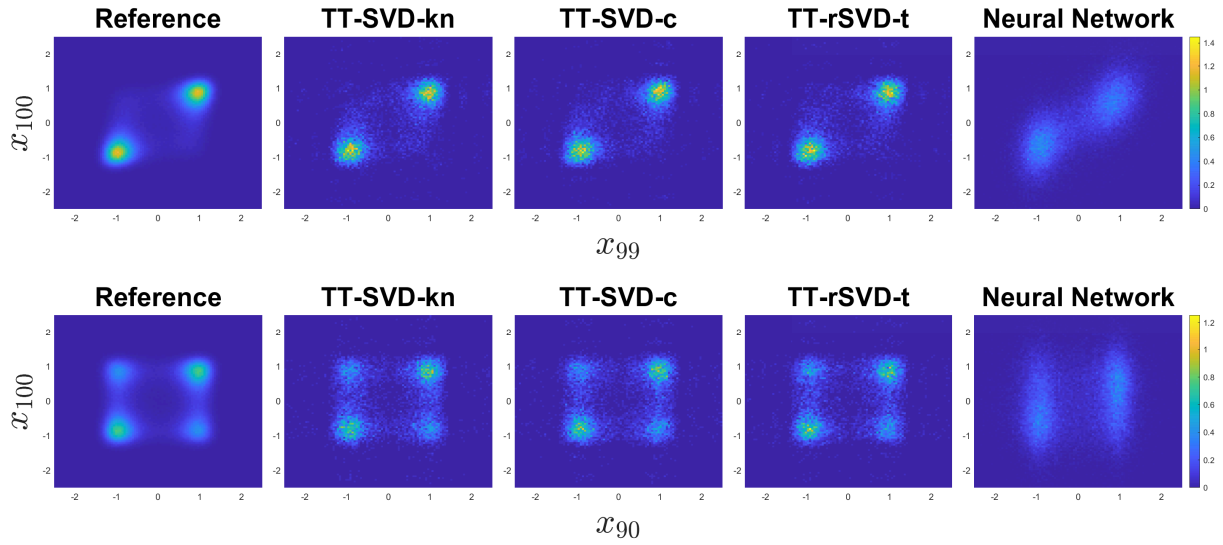
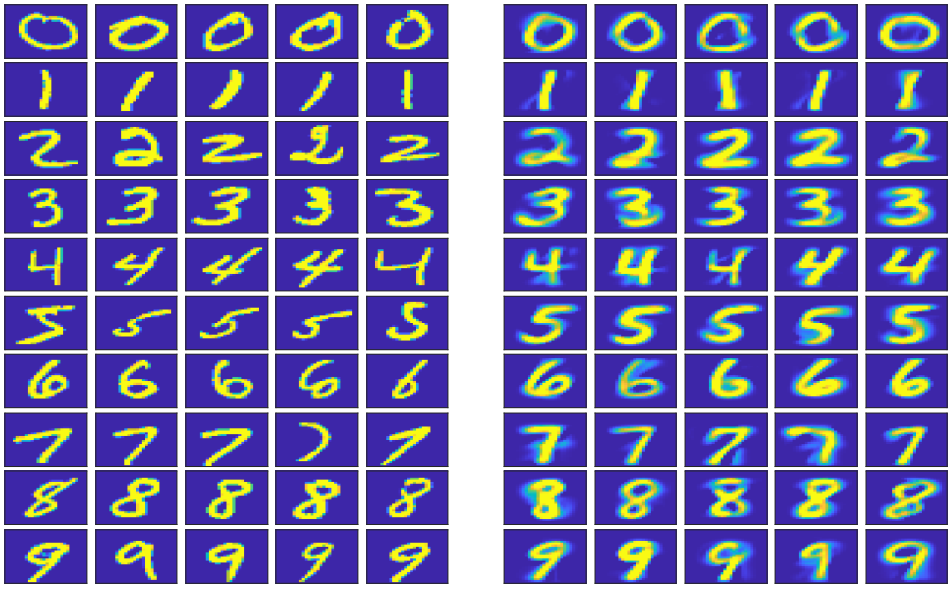


Figure 3.18: 1D G-L model for $d = 100$: visualization of (x_{99}, x_{100})-marginal (first row) and (x_{90}, x_{100})-marginal (second row) of samples from generative models. Left to right: reference samples, samples from tensor trains under three proposed algorithms, and samples from neural network.

where embedding dimension is set to $d' = 8$. After training, we independently sample five images for each digit from the learned TT, as shown in Figure 3.19(b). For comparison, Figure 3.19(a) displays five randomly selected images per digit from the original MNIST dataset. The generated images from the tensor train exhibit well-preserved digit structures and visually plausible variations across classes.

3.10 Conclusion

In this paper, we propose a natural approach to high-dimensional density estimation entirely based on linear algebra, which allows the construction of a density estimator as a low-rank functional tensor train. To this end, we provide four tensor decomposition algorithms. The first one is based on direct TT-SVD algorithm with fast implementation in a density estimation setting. The second algorithm follows a Nyström approximation. The third approach modifies the first approach by thresholding certain elements in the tensor and employing a hierarchical matrix factorization to accelerate computational efficiency. The last



(a) Real images (b) Generated images

Figure 3.19: Generated and real images from MNIST. For each digit class, five images are randomly selected for illustration.

algorithm employs a tensorized sketching technique. Several of these proposed algorithms can achieve linear computational complexity both in dimension d and in sample size N. Theoretical analysis demonstrates that tensor-train estimators can also achieve linear scaling of the estimation error with respect to the dimensionality d under some specific conditions. Comprehensive numerical experiments validate the accuracy of all proposed algorithms, in comparison to other density estimators based on neural networks.

CHAPTER 4

DENSITY ESTIMATION VIA HIERARCHICAL TENSOR SKETCHING

4.1 Introduction

Density estimation is one of the most fundamental problems in statistics and machine learning. Let p^* be a probability density function on a d-dimensional space $\mathbb{R}^d$. The task of density estimation is to estimate p^* based on a set of independently and identically distributed (i.i.d) samples $\{\boldsymbol{y}^i\}_{i=1}^N$ drawn from the density. More precisely, our goal is to estimate p^* from the sample empirical distribution,

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{y}^i}(x), \text{ where } x \in \mathbb{R}^d, \tag{4.1}$$

where $\delta_{\boldsymbol{y}^i}$ is the δ-measure supported on $\boldsymbol{y}^i \in \mathbb{R}^d$.

Traditional density estimators including the histograms [194, 144] and kernel density estimators [187, 173] (KDEs) typically perform well in low-dimensional settings. While it is possible to engineer kernels for the above methods that work in high-dimensional cases, recently it has been common to use neural network-based approaches that can learn features for high-dimensional density estimations. This includes autoregressive models [218, 69, 171], generative adversarial networks (GANs) [70], variational autoencoders (VAEs) [117], and normalizing flows [185, 14, 54, 76].

Alternatively, several works have proposed to approximate the density function in low-rank tensor networks, in particular the tensor train (TT) format [169] (known as matrix product state (MPS) in the physics literatures [178, 232]). Tensor-network represented distribution can efficiently generate independent and identically distributed (i.i.d.) samples by

applying conditional distribution sampling [55] to the obtained TT format. Furthermore, one can take advantage of the efficient TT format for fast downstream task applications, such as sampling, conditional sampling, computing the partition function, solving committor functions for large scale particle systems [36], etc.

In this work, we propose a randomized linear algebra based algorithm to estimate the probability density in the form of a hierarchical tensor-network given an empirical distribution, which further extends the MPS/TT [102] and tree tensor-network structure [210]. Hierarchical format applications play a critical role in statistical modeling by allowing for the representation of complex systems and the interactions between different local components. Various hierarchical structures have been proposed in the past with active applications in different fields, such as Bayesian modeling [126, 130], conditional generations [188], Gaussian process computations [34, 8, 67, 35], optimizations [199, 38, 106] and high-dimensional density modeling [66, 26]. The hierarchical structure can effectively handle spatial random field such as Ising models on lattice where the sites can be clustered hierarchically. Similar to [102], we use sketching technique to form a parallel system of linear equations with constant size with respect to the dimension d where the coefficients of the linear system are moments of the empirical distribution. By solving $d \log_2 d$ number of linear equations, one can obtain a hierarchical tensor approximation to the distribution.

Prior work

In the context of recovering low-rank tensor trains (TTs), there are two general types of input data. The first type involves the assumption that a d-dimensional function p can be evaluated at any point and it aims to recover p with a limited number of evaluations (typically polynomial in d). Techniques such as TT-cross [166], DMRG-cross [190], TT completion [207], and generalizations [227, 108] for tensor rings are under this category.

In the second scenario, where only samples from distribution are given, [21, 85, 161] propose a method to recover the density function by minimizing some measure of error, such as the Kullback-Leibler (KL) divergence, between the empirical distribution and the MPS/TT ansatz. However, due to the nonlinear parameterization of a TT in terms of the tensor components, optimization approaches can be prone to local minima issues. A recent work [102] proposes and analyzes an algorithm that outputs an MPS/TT representation of $\hat{p}$ to estimate p^*, by determining each MPS/TT core in parallel via a perspective that views randomized linear algebra routine as a method of moments.

Although our method shares some similarities with tensor compression techniques in the current literature and can be utilized for compressing a large exponentially sized tensor into a low-complexity hierarchical tensor, our focus is primarily on an estimation problem instead of an approximation problem. Specifically, we aim to estimate the ground truth distribution p^* that generates the empirical distribution $\hat{p}$ in terms of a TT, within a generative modeling context. Assuming the existence of an algorithm $\mathcal{A}$ that can take any d-dimensional function p and output its corresponding TT as $\mathcal{A}(p)$, then one would expect that such $\mathcal{A}$ could minimize the following differences:

$$p^* - \mathcal{A}(\hat{p}) = \underbrace{p^* - \mathcal{A}(p^*)}_{\text{approximation error}} + \underbrace{\mathcal{A}(p^*) - \mathcal{A}(\hat{p})}_{\text{estimation error}}$$

In generative modeling context, the empirical distribution $\hat{p}$ is affected by sample variance, leading to variance in the TT representation $\mathcal{A}(\hat{p})$ and, consequently, the estimation error. Our method is designed to minimize this estimation error.

In contrast, methods based on singular value decomposition (SVD) [169] and randomized linear algebra [166, 190, 197, 155] aim to compress the input function p as a TT such that $\mathcal{A}(p) \approx p$. Using such methods in the statistical learning context can result in an estimation error of $\mathcal{A}(p^*) - \mathcal{A}(\hat{p}) \approx p^* - \hat{p}$ which grows exponentially with the number of variables d

for a fixed number of samples.

Here we summarize our contributions.

1. We propose an optimization free method to construct a hierarchical tensor-network for approximating a high-dimensional probability density via empirical distribution in $O(Nd \log d)$ complexity. The proposed method is suitable for representing a distribution coming from a spatial random field.

2. We analyze the estimation error in terms of Frobenious norm for high-dimensional tensor. We show that the error decays like $O(\frac{c^{\log d}}{\sqrt{N}})$ for some constant c that depends on some easily computable norm of the reconstructed hierarchical tensor network.

The work is organized as follows. In the next section, we introduce notations and tensor diagrams in this paper, respectively. In Section 4.2, we present our algorithm for density estimation in terms of hierarchical tensor-network. In Section 4.3, we analyze the estimation error of the algorithm. In Section 4.4 we perform numerical experiments in several 1D and 2D Ising-like models. Finally, we conclude in Section 4.5.

Notations

For an integer $n \in \mathbb{N}$, we define $[n] = [1, 2, \cdots, n]$. For two integers $m, n \in \mathbb{N}$ where $n > m$, we use the `MATLAB` notation $m : n$ to denote the set $[m, m+1, \cdots, n]$.

In this paper, we will extensively work with vectors, matrices and tensors. We use boldface lower-case letters to denote vectors (*e.g.* $\boldsymbol{x}$). We denote matrices and tensors by capital letters (*e.g.* A). Let $\mathcal{I}$ and $\mathcal{J}$ be two sets of indices, we use $A(\mathcal{I}, \mathcal{J})$ to denote a submatrix of A with rows indexed by $\mathcal{I}$ and columns indexed by $\mathcal{J}$. We also use $A(\mathcal{I}, :)$ or $A(:, \mathcal{J})$ to denote a set of rows or columns, respectively. Similarly, we use $A(\mathcal{I}, :, :)$ to denote a set of first dimensional array indexed by $\mathcal{I}$ for a three-dimensional tensor A. To simplify the notation for matrix multiplication, we use : to represent the multiplied dimension. For instance, $A(\mathcal{I}, :)B(:, \mathcal{K}) = \sum_{j \in \mathcal{J}} A(\mathcal{I}, j)B(j, \mathcal{K})$.

Besides, we generalize the notion of Frobenius norm $\|\cdot\|_F$ such that for a d-dimensional tensor p, it is defined as

$$\|p\|_F = \sqrt{\sum_{x_1\in[n_1],\ldots,x_d\in[n_d]} p(x_1,\ldots,x_d)^2}. \tag{4.2}$$

When working with discrete high-dimensional functions, it is convenient to reshape the function into a matrix. We call these matrices as *unfolding matrices*. For a d-dimensional density $p:\ [n_1]\times[n_2]\times\cdots\times[n_d]\to\mathbb{R}$ for some $n_1,\cdots,n_d\in\mathbb{N}$, we define the k-th unfolding matrix of a d-dimensional function by $p(x_1,\cdots,x_k;x_{k+1},\cdots,x_d)$ or $p(x_{1:k},x_{k+1:d})$, which can be viewed as a matrix of shape $\prod_{i=1}^{k} n_i \times \prod_{i=k+1}^{d} n_i$.

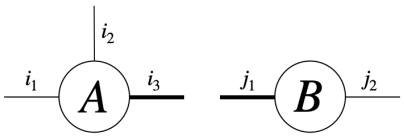
Tensor Diagrams

To illustrate the method, we use tensor diagram frequently in the paper. In tensor diagrams, a tensor is represented by a node, where the dimensionality of the tensor is indicated by the number of its incoming legs. We use circle to represent a node in the following tensor diagram. In Figure 4.1(A), a three-dimensional tensor A and a two-dimensional matrix B are depicted in the tensor diagram, which can be treated as two functions $A(i_1,i_2,i_3)$ and $B(j_1,j_2)$, respectively. Besides, we use bold line to illustrate that the specific dimension has a larger size compared to other dimensions. Thus the number of i_3 is essentially larger than the number of i_1 or i_2 and the same property holds for j_1 compared to j_2 in Figure 4.1(A).

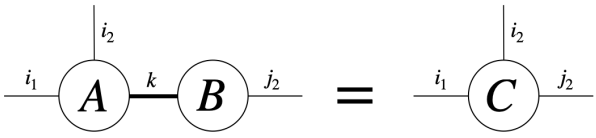
In addition, we could use tensor diagram to represent tensor multiplication or tensor contraction, which is able to provide a convenient way to understand this operation. In Figure 4.1(B), leg i_3 of A is joined with leg j_1 of B and this supposes implicitly that the two legs have the same size, represented by a new notation k. This corresponds to the computation: $\sum_k A(i_1,i_2,k)B(k,j_2) = C(i_1,i_2,j_2)$. Besides, reshaping is another significant operation of tensor. In Figure 4.1(C), we demonstrate reshaping by combining the first two legs of tensor

A (i_1 and i_2) into a new leg with a size equivalent to the product of the sizes of i_1 and i_2. This procedure converts a three-dimensional tensor into a two-dimensional matrix. This is illustrated in Figure 4.1(C). More specifically, to emphasize the large size of the combined dimensions, we use a bold line in the right diagram of Figure 4.1(C), with index $i_{1:2}$, which follows the notation introduced before.

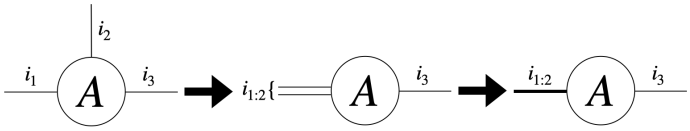
Further background about tensor diagram and tensor notations could be found in [163]. An example is depicted in Figure 4.3 and this would be explained in detail in the next section.



(a) Tensor diagrams of 3-dimensional A and 2-dimensional B.



(b) Tensor diagram of tensor multiplication.



(c) Tensor diagrams of reshaping.

Figure 4.1: Tensor diagrams examples.

4.2 Density Estimation via Hierarchical Tensor-network

In this section, we estimate a discrete d-dimensional density p from the empirical distribution in terms of a hierarchical tensor-network. For convenience, we assume $d = 2^L$, and we partition the dimension in a binary fashion to construct our hierarchical tensor-network, although the proposed method can be easily generalized to arbitrary choice of d. We assume $p : [n_1] \times [n_2] \times \cdots \times [n_d] \to \mathbb{R}$ for some $n_1, \cdots, n_d \in \mathbb{N}$. Given N i.i.d samples from the density p, our objective is to approximate the density using a hierarchical tensor representation.

4.2.1 Main Idea

We start with a simple example to illustrate the main idea. Suppose a density only has two variables $p(x_1, x_2)$ and $\text{rank}(p) = r$. We can obtain a decomposition of p via the following procedure: We first determine the column and row spaces of p as $p_1(x_1, \gamma_1)$ and $p_2(x_2, \gamma_2)$, $\gamma_1, \gamma_2 \in [r]$, then we determine a matrix $G \in \mathbb{R}^{r\times r}$ via solving the following equation for all x_1, x_2:

$$p_1(x_1, :)Gp_2(x_2, :)^T = p(x_1, x_2), \tag{4.3}$$

which in turn provides a low-rank approximation to p. For high-dimensional cases, we can recursively apply this binary partition to determine a hierarchically low-rank tensor-network.

More specifically, let $l = 0, \ldots, L$ be the levels of the hierarchical tensor-network. At each level l, we cluster the dimensions $[d]$ into 2^l parts and $k \in [2^l]$ is the index of nodes per level. Clusters of each level are defined as $\mathcal{C}_k^{(l)} := (k-1)(\frac{2^L}{2^l}) + 1 : k(\frac{2^L}{2^l})$ for $k \in [2^l]$, and thus they partition $[d]$. The main idea is to solve for the nodes $G_k^{(l)}$ at each level via a set of core defining equations:

$$\begin{aligned}
p_1^{(1)}(\mathcal{I}_1^{(1)}, :)G_1^{(0)}(:, :, \tilde{\gamma}_1^{(0)})p_2^{(1)}(\mathcal{I}_2^{(1)}, :)^T &= p_1^{(0)}(\mathcal{I}_1^{(1)}, \mathcal{I}_2^{(1)}, \tilde{\gamma}_1^{(0)}),\ \tilde{\gamma}_1^{(0)} \in [1],\\
p_{2k-1}^{(2)}(\mathcal{I}_{2k-1}^{(2)}, :)G_k^{(1)}(:, :, \tilde{\gamma}_k^{(1)})p_{2k}^{(2)}(\mathcal{I}_{2k}^{(2)}, :)^T &= p_k^{(1)}(\mathcal{I}_{2k-1}^{(2)}, \mathcal{I}_{2k}^{(2)}, \tilde{\gamma}_k^{(1)}), \tilde{\gamma}_k^{(1)} \in [\tilde{r}^{(1)}], k \in [2],\\
&\vdots\\
p_{2k-1}^{(l)}(\mathcal{I}_{2k-1}^{(l)}, :)G_k^{(l-1)}(:, :, \tilde{\gamma}_k^{(l-1)})p_{2k}^{(l)}(\mathcal{I}_{2k}^{(l)}, :)^T &= p_k^{(l-1)}(\mathcal{I}_{2k-1}^{(l)}, \mathcal{I}_{2k}^{(l)}, \tilde{\gamma}_k^{(l-1)}), \tilde{\gamma}_k^{(l-1)} \in [\tilde{r}^{(l-1)}], k \in [2^{l-1}],\\
&\vdots\\
p_{2k-1}^{(L)}(\mathcal{I}_{2k-1}^{(L)}, :)G_k^{(L-1)}(:, :, \tilde{\gamma}_k^{(L-1)})p_{2k}^{(L)}(\mathcal{I}_{2k}^{(L)}, :)^T &= p_k^{(L-1)}(\mathcal{I}_{2k-1}^{(L)}, \mathcal{I}_{2k}^{(L)}, \tilde{\gamma}_k^{(L-1)}), \tilde{\gamma}_k^{(L-1)} \in [\tilde{r}^{(L-1)}], k \in [2^{L-1}],\\
G_k^{(L)} &= p_k^{(L)}, k \in [2^L].
\end{aligned} \tag{4.4}$$

The above equations (4.4) are the key equations in this paper. Here $p_1^{(0)} := p$ and $\mathcal{I}_k^{(l)} = \prod_{i\in\mathcal{C}_k^{(l)}}[n_i]$ is the set of all possible values for $x_{\mathcal{C}_k^{(l)}}$ for $k \in [2^l], l \in [L]$. Similarly, we

denote $\mathcal{J}_k^{(l)} = \prod_{i\in[d]\setminus\mathcal{C}_k^{(l)}}[n_i]$ as the set of all possible values for $x_{[d]\setminus\mathcal{C}_k^{(l)}}$ for every k, l. For convenience, we denote $G_k^{(L)} := p_k^{(L)}$ for $k \in [2^L]$ in the last level. The core equations could be depicted directly via the tensor diagram in Figure 4.2, following the pattern in tensor diagram Section. Here clusters $\mathcal{C}_k^{(l)}$ satisfy the following relationship between two levels: $\mathcal{C}_k^{(l)} = \mathcal{C}_{2k-1}^{(l+1)} \cup \mathcal{C}_{2k}^{(l+1)}$, and furthermore, $\mathcal{I}_k^{(l)} = \mathcal{I}_{2k-1}^{(l+1)} \times \mathcal{I}_{2k}^{(l+1)}$ holds for $k \in [2^l], l \in [L-1]$. Then $p_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\gamma}_k^{(l)}), \tilde{\gamma}_k^{(l)} \in [\tilde{r}^{(l)}]$ is obtained by reshaping the first two indices of $p_k^{(l)}(\mathcal{I}_{2k-1}^{(l+1)}, \mathcal{I}_{2k}^{(l+1)}, \tilde{\gamma}_k^{(l)})$. We use $p_k^{(l)}$ to represent the two-dimensional form unless we emphasize the third index of it.

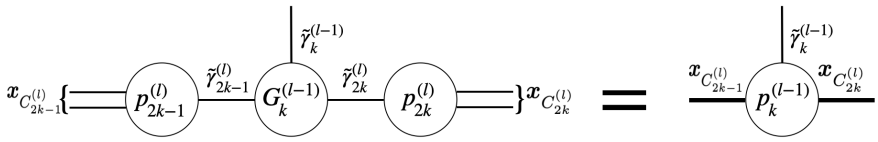


Figure 4.2: Tensor diagram of core defining equations (4.4). In order to go from left hand side to right hand side, we use the reshaping operation in Figure 4.1(C) to reshape the dimension $(x_{\mathcal{C}_{4k-3}^{(l+1)}}, x_{\mathcal{C}_{4k-2}^{(l+1)}})$ of $p_{2k-1}^{(l)}$ into $x_{\mathcal{C}_{2k-1}^{(l)}}$ and reshape the dimension $(x_{\mathcal{C}_{4k-1}^{(l+1)}}, x_{\mathcal{C}_{4k}^{(l+1)}})$ of $p_{2k}^{(l)}$ into $x_{\mathcal{C}_{2k}^{(l)}}$.

We now state an assumption for the unfolding density $p_k^{(l)}$ that guarantees that each equation in (4.4) would have solution $G_k^{(l)}$.

Assumption 5. *We assume* $\text{Range}\left(p_{2k-1}^{(l)}\right) \supset \text{Range}\left(p_k^{(l-1)}(\mathcal{I}_{2k-1}^{(l)}; \mathcal{I}_{2k}^{(l)}, \tilde{\gamma}_k^{(l-1)})\right)$ *and* $\text{Range}\left(p_{2k}^{(l)}\right) \supset \text{Range}\left(p_k^{(l-1)}(\mathcal{I}_{2k}^{(l)}; \mathcal{I}_{2k-1}^{(l)}, \tilde{\gamma}_k^{(l-1)})\right)$, *respectively, for* $k \in [2^{l-1}]$ *and* $l \in [L]$.

Assumption 5 in turn gives us the following theorem, which guarantees a hierarchical tensor-network representation of p in terms of $\{G_k^{(l)}\}_{k,l}$, as depicted by a tensor diagram in Figure 4.3.

Theorem 5. *Suppose Assumption 5 holds, then* $\{\{G_k^{(l)}\}_{k=1}^{2^l}\}_{l=0}^{L}$ *in* (4.4) *gives a hierarchical tensor-network representation of* p.

Proof. Based on Assumption 5, we have $\text{Range}(p_1^{(0)}(\mathcal{I}_1^{(1)}; \mathcal{I}_2^{(1)}, \tilde{\gamma}_1^{(0)})) \subset \text{Range}(p_1^{(1)})$ and $\text{Range}(p_1^{(0)}(\mathcal{I}_2^{(1)}; \mathcal{I}_1^{(1)}, \tilde{\gamma}_1^{(0)})) \subset \text{Range}(p_2^{(1)})$, therefore the equation defining $G_1^{(0)}$ in (4.4)

admits a solution, which comes from taking the pseudo-inverse of $p_1^{(1)}, p_2^{(1)}$, i.e.

$$G_1^{(0)}(:,:,\tilde{\gamma}_1^{(0)}) = \left(p_1^{(1)}\right)^\dagger p_1^{(0)}(:,:,\tilde{\gamma}_1^{(0)}) \left(p_2^{(1)T}\right)^\dagger, \tilde{\gamma}_1^{(0)} \in [1],$$

Now we already construct a representation of $p_1^{(0)} = p$ based on $p_1^{(1)}, p_2^{(1)}$ and $G_1^{(0)}$, after solving the equation involving $G_1^{(0)}$ in (4.4). Furthermore, we could express $p_1^{(1)}$ and $p_2^{(1)}$ in terms of $G_1^{(1)}, p_1^{(2)}, p_2^{(2)}$ and $G_2^{(1)}, p_3^{(2)}, p_4^{(2)}$ by solving the equations for $G_1^{(1)}$ and $G_2^{(1)}$ in (4.4) (which again admits solutions due to Assumption 5). By recursing this procedure on $p_1^{(2)}, p_2^{(2)}, p_3^{(2)}, p_4^{(2)}$, and the lower level $\{\{p_k^{(l)}\}_{k=1}^{2^l}\}_{l=3}^{L}$, we obtain a tensor-network representation of p in terms of $\{\{G_k^{(l)}\}_{k=1}^{2^l}\}_{l=0}^{L}$.

□

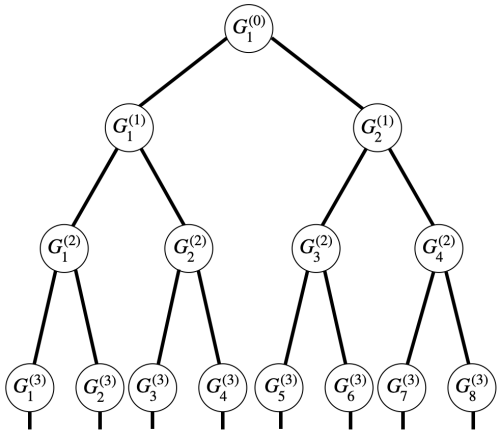


Figure 4.3: Tensor diagram of the hierarchical tensor-network representing an 8-dimensional density p.

At this point there are two issues that need to be addressed in order to use (4.4) to determine the tensor cores $\{G_k^{(l)}\}_{k,l}$. The first issue is on how to obtain the range $p_k^{(l)}$ at each level. The second issue is that while one can determine the cores $\{G_k^{(l)}\}_{k,l}$ via (4.4), in practice it is challenging to estimate the exponential sized coefficient matrix $\{p_k^{(l)}\}_{k,l}$ (More specifically, it scales as $|\mathcal{I}_k^{(l)}|$, exponential with respect to $|\mathcal{C}_k^{(l)}|$) via only finite number of samples. To this end, we need to reduce the size of the system in (4.4).

We deal with these two issues as follows:

- **Obtaining** $p_k^{(l)}$: Inspired by how one "sketches" the range of a matrix in randomized linear algebra literature [166, 190], we define

$$p_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\gamma}_k^{(l)}) = p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)}) T_k^{(l)}(\mathcal{J}_k^{(l)}, \tilde{\gamma}_k^{(l)}), \quad \tilde{\gamma}_k^{(l)} \in [\tilde{r}^{(l)}], \tag{4.5}$$

  which has size $|\mathcal{I}_k^{(l)}| \times \tilde{r}^{(l)}$ and sketches the range of matrix $p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$ via multiplying it with sketch function $T_k^{(l)}(\mathcal{J}_k^{(l)}, \tilde{\gamma}_k^{(l)})$. It is desirable if the sketch function can capture the range: $\text{Range}(p_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\gamma}_k^{(l)})) = \text{Range}(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)}) T_k^{(l)}(\mathcal{J}_k^{(l)}, \tilde{\gamma}_k^{(l)})) = \text{Range}(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)}))$. The choice of sketch function will be discussed in Section 4.2.3.

- **Reducing the number of equations**: Note that the system of equations in (4.4) is over-determined, as each $G_k^{(l)}(:, :, \tilde{\gamma}_k^{(l)})$ is of size at most $\tilde{r}^{(l+1)} \times \tilde{r}^{(l+1)}$. Therefore in principle, one can reduce the rows and columns of each equation in (4.4) such that each equation involves only $\tilde{r}^{(l+1)} \times \tilde{r}^{(l+1)}$ coefficient matrices. This can be done by applying sketch function yet another time. In particular, we apply $S_{2k-1}^{(l)}(\mathcal{I}_{2k-1}^{(l)}, \tilde{\nu}_{2k-1}^{(l)})$ and $S_{2k}^{(l)}(\mathcal{I}_{2k}^{(l)}, \tilde{\nu}_{2k}^{(l)})$, $\tilde{\nu}_{2k-1}^{(l)}, \tilde{\nu}_{2k}^{(l)} \in [\tilde{r}^{(l)}]$ to both sides of the equations in (4.4) which results in the following equations:

$$A_{2k-1}^{(l)}(\tilde{\nu}_{2k-1}^{(l)}, :) G_k^{(l-1)}(:, :, \tilde{\gamma}_k^{(l-1)}) A_{2k}^{(l)}(\tilde{\nu}_{2k}^{(l)}, :)^T = B_k^{(l-1)}(\tilde{\nu}_{2k-1}^{(l)}, \tilde{\nu}_{2k}^{(l)}, \tilde{\gamma}_k^{(l-1)}), \tilde{\gamma}_k^{(l-1)} \in [\tilde{r}^{(l-1)}], \tag{4.6}$$

  for $k \in [2^{l-1}], l \in [L]$ where

$$A_k^{(l)}(\tilde{\nu}_k^{(l)}, \tilde{\gamma}_k^{(l)}) = S_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\nu}_k^{(l)})^T p_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\gamma}_k^{(l)}) = S_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\nu}_k^{(l)})^T p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)}) T_k^{(l)}(\mathcal{J}_k^{(l)}, \tilde{\gamma}_k^{(l)}), \tag{4.7}$$

  and

$$B_k^{(l)}(\tilde{\nu}_{2k-1}^{(l+1)}, \tilde{\nu}_{2k}^{(l+1)}, \tilde{\gamma}_k^{(l)}) = S_{2k-1}^{(l+1)}(\mathcal{I}_{2k-1}^{(l+1)}, \tilde{\nu}_{2k-1}^{(l+1)})^T p_k^{(l)}(\mathcal{I}_{2k-1}^{(l+1)}, \mathcal{I}_{2k}^{(l+1)}, \tilde{\gamma}_k^{(l)}) S_{2k}^{(l+1)}(\mathcal{I}_{2k}^{(l+1)}, \tilde{\nu}_{2k}^{(l+1)}). \tag{4.8}$$

At this point, one can solve for $G_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)})$ in (4.6) via pseudo-inverse

$$G_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)}) = (A_{2k-1}^{(l)})^\dagger B_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)})(A_{2k}^{(l)T})^\dagger, k \in [2^{l-1}], l \in [L],$$
$$G_k^{(L)}(\mathcal{I}_k^{(L)}, \tilde{\gamma}_k^{(L)}) = p_k^{(L)}(\mathcal{I}_k^{(L)}, \tilde{\gamma}_k^{(L)}), k \in [2^L]. \tag{4.9}$$

The equations with reduced size could be shown via the tensor diagram in Figure 4.4:

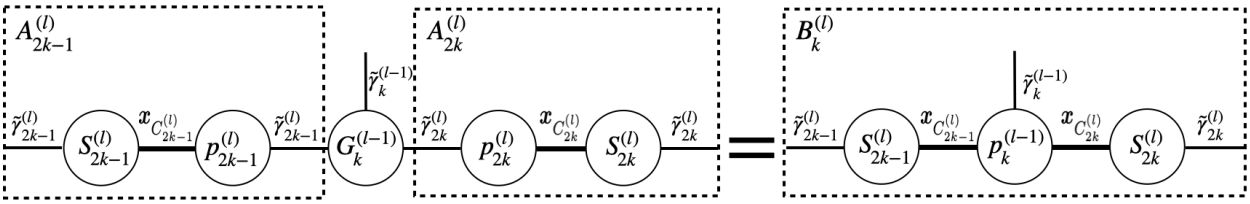


Figure 4.4: Tensor diagram of reduced core defining equations (4.6).

The following theorem guarantees that the solution (4.9) to the reduced system of equations gives a tensor-network representation of p.

Theorem 6. *If* $\text{Range}\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})T_k^{(l)}\right) = \text{Range}\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})\right)$ *and* $\text{Range}\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T S_k^{(l)}\right) = \text{Range}\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T\right)$ *for* $k \in [2^l], l \in [L]$, *then* $\{\{G_k^{(l)}\}_{k=1}^{2^l}\}_{l=0}^{L}$ *defined in* (4.9) *gives a hierarchical tensor-network representation of* p.

Proof. We first show that $\{G_k^{(l)}\}$ in (4.9) gives a solution for (4.4), with the definition of $p_k^{(l)}$ in (4.5): $p_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\gamma}_k^{(l)}) = p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})T_k^{(l)}(\mathcal{J}_k^{(l)}, \tilde{\gamma}_k^{(l)})$. This could help to establish (4.9) as a tensor-network representation of p via Theorem 5.

To begin with, we show $\text{Range}\left(A_k^{(l)T}\right) = \text{Range}\left(p_k^{(l)T}\right)$ for every k, l, indicating that the row spaces of $A_k^{(l)}$ and $p_k^{(l)}$ are the same. This is due to the fact that $\text{Range}\left(p_k^{(l)T}\right) = \text{Range}\left(T_k^{(l)T} p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T\right) = \text{Range}\left(T_k^{(l)T} p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T S_k^{(l)}\right) = \text{Range}\left(A_k^{(l)T}\right)$. The first and last equalities are based on the definition of $p_k^{(l)}$ and $A_k^{(l)}$, and second equality is due to the assumption of the statement where $\text{Range}\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T S_k^{(l)}\right) = \text{Range}\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T\right)$

and applying $T_k^{(l)T}$ to the row space does not change the range. Thus $G_k^{(l-1)}$ defined in (4.9) satisfies equality $p_{2k-1}^{(l)} G_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)}) p_{2k}^{(l)T} = p_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)})$ for $\tilde{\gamma}_k^{(l-1)} \in [\tilde{r}^{(l-1)}]$.

We have shown that $\{G_k^{(l)}\}$ in (4.9) is a solution to (4.4) when $\{p_k^{(l)}\}$ is defined in (4.5). The proof can be concluded if we further show $\{p_k^{(l)}\}$ in (4.5) satisfies Assumption 5, hence Theorem 5 can be directly applied to show $\{G_k^{(l)}\}$ gives a representation of p. For a specific choice of k and l, Range $\left(p_{2k-1}^{(l)}\right)$ = Range $\left(p(\mathcal{I}_{2k-1}^{(l)}; \mathcal{J}_{2k-1}^{(l)}) T_{2k-1}^{(l)}\right)$ = Range $\left(p(\mathcal{I}_{2k-1}^{(l)}; \mathcal{J}_{2k-1}^{(l)})\right)$ =
Range $\left(p(\mathcal{I}_{2k-1}^{(l)}; \mathcal{I}_{2k}^{(l)}, \mathcal{J}_k^{(l-1)})\right) \supset$ Range $\left(p(\mathcal{I}_{2k-1}^{(l)}; \mathcal{I}_{2k}^{(l)}, \tilde{\gamma}_k^{(l-1)})\right)$ where the first equality is by definition (4.5), second equality is due to the assumption of the theorem, the third equality is due to the fact that $\mathcal{J}_{2k-1}^{(l)} = \mathcal{I}_{2k}^{(l)} \times \mathcal{J}_k^{(l-1)}$ and the first inclusion holds since $p(\mathcal{I}_{2k-1}^{(l)}; \mathcal{I}_{2k}^{(l)}, \tilde{\gamma}_k^{(l-1)})$ is formed from applying the sketch function $T_k^{(l-1)}$ to the columns of $p(\mathcal{I}_{2k-1}^{(l)}; \mathcal{I}_{2k}^{(l)}, \mathcal{J}_k^{(l-1)})$. Similarly, Range$(p_{2k}^{(l)}) \supset$ Range $\left(p(\mathcal{I}_{2k}^{(l)}; \mathcal{I}_{2k-1}^{(l)}, \tilde{\gamma}_k^{(l-1)})\right)$, thus Assumption 5 is satisfied. □

Now, $\tilde{r}^{(l)}, l = 1, \ldots, L$ depends on the number of sketch functions chosen. As we tend to choose a large set of sketch functions in order to capture the range of $p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$ and $p_k^{(l)T}$ properly, this gives rise to three issues: (1) from a numerical standpoint, one may run into stability issue when taking the pseudo-inverse of a nearly low rank matrix. If $p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$ is nearly rank $r^{(l)} \leq \tilde{r}^{(l)}$, then one would expect $A_k^{(l)}(\tilde{\nu}_k^{(l)}, \tilde{\gamma}_k^{(l)}) = S_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\nu}_k^{(l)})^T p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)}) T_k^{(l)}(\mathcal{J}_k^{(l)}, \tilde{\gamma}_k^{(l)})$ to be also nearly rank $r^{(l)} \leq \tilde{r}^{(l)}$. Taking the pseudo-inverse of $A_k^{(l)}$ can cause instability. (2) From a statistical standpoint, it is challenging to estimate $\{A_k^{(l)}\}_{k,l}$ and $\{B_k^{(l)}\}_{k,l}$, the moments of p, from a fixed number of samples in $\hat{p}$ when $\tilde{r}^{(l)}$ is large. (3) From a computational viewpoint, large $\tilde{r}^{(l)}$ causes large $G_k^{(l)}$, leading to high complexity in storing and deploying the hierarchical tensor network. In order to solve these issues caused by over-sketching, we introduce a trimming strategy in Section 4.2.2.

4.2.2 Trimming

In practice, we often apply a low-rank truncation to each $A_k^{(l)}$. Let $A_k^{(l)} \to \mathcal{P}_{r^{(l)}}(A_k^{(l)})$, where $\mathcal{P}_{r^{(l)}}(\cdot)$ provides the best rank-$r^{(l)}$ approximation to a matrix in terms of the spectral norm. Then we could solve

$$\mathcal{P}_{r^{(l)}}(A_{2k-1}^{(l)})G_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)})\mathcal{P}_{r^{(l)}}(A_{2k}^{(l)})^T = B_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)}), \tilde{\gamma}_k^{(l-1)} \in [\tilde{r}^{(l-1)}], k \in [2^{l-1}], l \in [L] \tag{4.10}$$

instead of (4.6). The following proposition shows why this could be feasible:

Proposition 1. *Under the assumption of Theorem 6, if rank* $\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})\right) = r^{(l)} \leq \tilde{r}^{(l)}$ *for* $k \in [2^l], l \in [L]$, *then* $\mathcal{P}_{r^{(l)}}(A_k^{(l)}) = A_k^{(l)}$.

Proof. For every $k \in [2^l], l \in [L]$, since Range $\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})T_k^{(l)}\right)$ = Range $\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})\right)$ and rank $\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})\right) = r^{(l)} \leq \tilde{r}^{(l)}$, then rank $\left(T_k^{(l)}\right) = r^{(l)}$. Similarly, rank $\left(S_k^{(l)}\right) = r^{(l)}$ based on Range $\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T S_k^{(l)}\right)$ = Range $\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})^T\right)$. Furthermore, $A_k^{(l)} = S_k^{(l)}(\mathcal{I}_k^{(l)}, :)^T p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$
$T_k^{(l)}(\mathcal{J}_k^{(l)}, :) \in \mathbb{R}^{\tilde{r}^{(l)} \times \tilde{r}^{(l)}}$ in (4.7) is also of rank $r^{(l)}$. Therefore, $\mathcal{P}_{r^{(l)}}(A_k^{(l)}) = A_k^{(l)}$ due to the fact that $\mathcal{P}_{r^{(l)}}(\cdot)$ provides the best rank-$r^{(l)}$ approximation. □

For $A_k^{(l)}$ with exactly rank $r^{(l)}$, the projection $\mathcal{P}_{r^{(l)}}(\cdot)$ in (4.10) seems vacuous following Proposition 1. However, this becomes crucial when $A_k^{(l)}$ is estimated from empirical moments, whose rank is higher than $r^{(l)}$, as mentioned in the end of last subsection. Now the solution becomes

$$G_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)}) = \left(\mathcal{P}_{r^{(l)}}(A_{2k-1}^{(l)})\right)^\dagger B_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)}) \left(\mathcal{P}_{r^{(l)}}(A_{2k}^{(l)})^T\right)^\dagger, \tilde{\gamma}_k^{(l-1)} \in [\tilde{r}^{(l-1)}], \tag{4.11}$$

for $k \in [2^{l-1}], l \in [L]$. Let the right singular matrices of $\mathcal{P}_{r^{(l)}}(A_{2k-1}^{(l)})$ and $\mathcal{P}_{r^{(l)}}(A_{2k}^{(l)})$ be $V_{2k-1}^{(l)}, V_{2k}^{(l)} \in \mathbb{R}^{\tilde{r}^{(l)} \times r^{(l)}}$. Since the range of $G_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)})$ is given by $V_{2k-1}^{(l)}, V_{2k}^{(l)}$, one

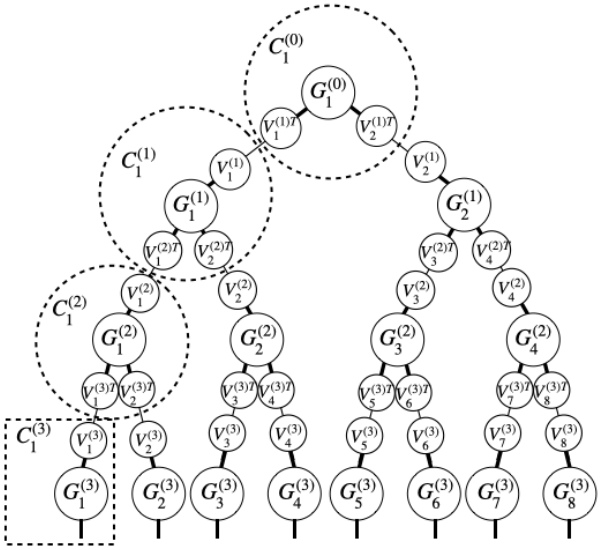
can have a reduced size tensor-network with rank $\{r^{(l)}\}_l$ via defining

$$C_k^{(l-1)}(:,:,\gamma_k^{(l-1)}) = \sum_{\tilde{\gamma}_k^{(l-1)} \in [\tilde{r}^{(l-1)}]} \left(V_{2k-1}^{(l)T} G_k^{(l-1)}(:,:,\tilde{\gamma}_k^{(l-1)}) V_{2k}^{(l)} \right) V_k^{(l-1)}(\tilde{\gamma}_k^{(l-1)}, \gamma_k^{(l-1)}), \gamma_k^{(l-1)} \in [r^{(l-1)}], \tag{4.12}$$

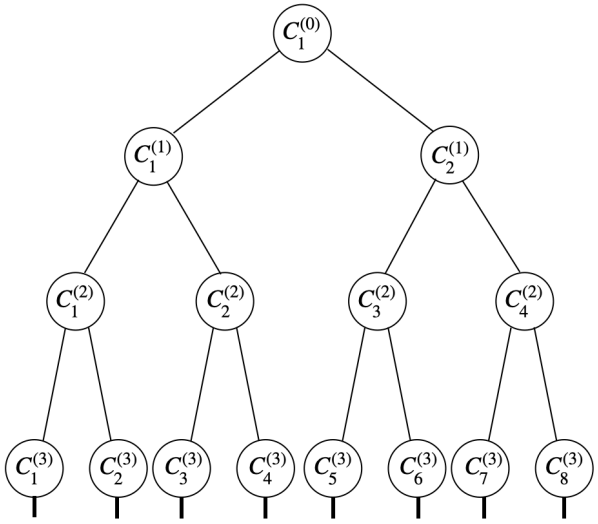
for $k \in [2^{l-1}], l \in [L]$. Besides, we have a similar definition of $C_k^{(L)}$ in the last level:

$$C_k^{(L)}(\mathcal{I}_k^{(l)}, \gamma_k^{(l)}) = G_k^{(L)}(\mathcal{I}_k^{(L)}, :) V_k^{(L)}(:, \gamma_k^{(L)}), k \in [2^L]. \tag{4.13}$$

The set of nodes $\{\{C_k^{(l)}\}_{k=1}^{2^l}\}_{l=0}^{L}$ gives rise to a hierarchical tensor-network in Figure 4.5, via inserting $V_{2k-1}^{(l)} V_{2k-1}^{(l)T}, V_{2k}^{(l)} V_{2k}^{(l)T}$ for the corresponding $G_k^{(l-1)}$ in the hierarchical tensor-network and regrouping the components, as shown in Figure 4.5(A). To highlight that the smaller size of $C_k^{(l)}$ compared to $G_k^{(l)}$, we represent legs of $C_k^{(l)}$ with thin lines and legs of $G_k^{(l)}$ with bold lines. In the following sections, we use this trimmed hierarchical tensor-network in Figure 4.5(B) as a representation of probability p.



(a) Insert $V_k^{(l)} V_k^{(l)T}$ and regroup the components in the hierarchical tensor-network.



(b) Trimmed hierarchical tensor-network from simplifying the representation with new variables.

Figure 4.5: Tensor diagram of trimmed hierarchical tensor-network.

4.2.3 Choice of Sketch Function

In this subsection, we introduce the choice of sketch function $\{S_k^{(l)}\}_{k,l}$ and $\{T_k^{(l)}\}_{k,l}$. There are two criteria for such a selection. (1) Since each $A_k^{(l)}$ is a collection of moments, one needs to be able to efficiently estimate $A_k^{(l)}$ from the empirical distribution $\hat{p}$. (2) $T_k^{(l)}$ needs to capture the range of $p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$ and furthermore, $S_k^{(l)}$ needs to capture the row space of $p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$, as the assumption in Theorem 6.

We deal with the choice of basis via using a *cluster* basis, which is successfully implemented in atomic cluster expansion [58, 61]. Starting from a single variable basis set $\{\phi_i\}_{i=1}^n$, we construct the cluster set of one variable:

$$\mathbf{B}^{1,d} := \bigcup_{k_1=1}^{d} \left(\bigcup_{i_{k_1}=1}^{n} \{\phi_{i_{k_1}}(x_{k_1})\} \right), \tag{4.14}$$

and cluster set of two variables:

$$\mathbf{B}^{2,d} := \bigcup_{(k_1,k_2)\in\binom{[d]}{2}} \left(\bigcup_{i_{k_1},i_{k_2}=1}^{n} \{\phi_{i_{k_1}}(x_{k_1})\phi_{i_{k_2}}(x_{k_2})\} \right), \tag{4.15}$$

and cluster set of t variables in general:

$$\mathbf{B}^{t,d} := \bigcup_{(k_1,\ldots,k_t)\in\binom{[d]}{t}} \left(\bigcup_{i_{k_1},\cdots,i_{k_t}=1}^{n} \{\phi_{i_{k_1}}(x_{k_1})\cdots\phi_{i_{k_t}}(x_{k_t})\} \right). \tag{4.16}$$

Commonly, $\{\phi_i\}_{i=1}^n$ are chosen as a set of orthonormal basis functions for convenience.

The way that we apply the sketching function is straightforward. We define

$$S_k^{(l)}(\cdot, \tilde{\nu}_k^{(l)}) \in \mathbf{B}^{t,|\mathcal{C}_k^{(l)}|}, \quad T_k^{(l)}(\cdot, \tilde{\gamma}_k^{(l)}) \in \mathbf{B}^{t,d-|\mathcal{C}_k^{(l)}|}, \tag{4.17}$$

where clusters $\mathcal{C}_k^{(l)} = (k-1)(\frac{2^L}{2^l}) + 1 : k(\frac{2^L}{2^l})$ for $k \in [2^l], l \in [L]$, as defined previously

in Section 4.2.1. Therefore, to obtain $A_k^{(l)}$ in (4.7), we evaluate $S_k^{(l)}(\cdot, \tilde{\nu}_k^{(l)}) : x_{\mathcal{C}_k^{(l)}} \to \mathbb{R}$ for all possible choices of $x_{\mathcal{C}_k^{(l)}}$ to obtain the sketch matrix $S_k^{(l)}(\mathcal{I}_k^{(l)}, \tilde{\nu}_k^{(l)})$, and evaluate $T_k^{(l)}(\cdot, \tilde{\gamma}_k^{(l)}) : x_{[d]\setminus\mathcal{C}_k^{(l)}} \to \mathbb{R}$ for all possible choices of $x_{[d]\setminus\mathcal{C}_k^{(l)}}$ to obtain the sketch matrix $T_k^{(l)}(\mathcal{J}_k^{(l)}, \tilde{\gamma}_k^{(l)})$. In practice, when we estimate $A_k^{(l)}$ from empirical distribution $\hat{p}$ we do not need to consider all possible values of $x_{\mathcal{C}_k^{(l)}}, x_{[d]\setminus\mathcal{C}_k^{(l)}}$ when forming $A_k^{(l)}$, but only those in the support of $\hat{p}$. Therefore, we only need to evaluate the sketch functions N times. We again use $S_{2k-1}^{(l)}(\mathcal{I}_{2k-1}^{(l)}, \tilde{\nu}_{2k-1}^{(l)}), S_{2k}^{(l)}(\mathcal{I}_{2k}^{(l)}, \tilde{\nu}_{2k}^{(l)})$ and $T_k^{(l-1)}(\mathcal{J}_k^{(l-1)}, \tilde{\gamma}_k^{(l-1)})$ together with $p(\mathcal{I}_{2k-1}^{(l)}, \mathcal{I}_{2k}^{(l)}, \mathcal{J}_k^{(l-1)})$ to obtain $B_k^{(l-1)}$ in (4.8).

Under this construction, sketch index set $\{\tilde{\nu}_k^{(l)}\}$ has cardinality $\binom{|\mathcal{C}_k^{(l)}|}{t} n^t$, significantly larger than true rank $r^{(l)}$ of $p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$, especially in the case when cluster size $|\mathcal{C}_k^{(l)}|$ is large. As a result, the sketch may be substantially oversized, which can in turn create difficulties in the subsequent trimming step.

To solve above issue, we develop randomized cluster basis following randomized singular value decomposition technique [82]. For our problem, we assume that rank $\left(p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})\right) = r^{(l)}$ is small, so a sketch with only a modest number of columns is sufficient to capture the relevant column space. Let $\tilde{S}_k^{(l)}(\cdot, \cdot) \in \mathbf{B}^{t, |\mathcal{C}_k^{(l)}|}$ denote the original cluster basis. We define the randomized basis

$$S_k^{(l)}(x_{\mathcal{C}_k^{(l)}}, \tilde{\nu}_k^{(l)}) = \tilde{S}_k^{(l)}(x_{\mathcal{C}_k^{(l)}}, :) W(:, \tilde{\nu}_k^{(l)}), \tilde{\nu}_k^{(l)} \in [\tilde{r}^{(l)}], \tag{4.18}$$

where $W \in \mathbb{R}^{\binom{|\mathcal{C}_k^{(l)}|}{t} n^t \times \tilde{r}^{(l)}}$ is a random matrix and we assume that it has orthonormal columns for convenience: $\langle W(:, i), W(:, j) \rangle = \delta_{i,j}$. Here we could choose $\tilde{r}^{(l)}$, a dimension-independent constant, slightly larger than true rank $r^{(l)}$ of $p(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)})$, to avoid oversketching issue generated from original cluster basis. Besides, we assume cluster basis matrix $\tilde{S}_k^{(l)}$ have orthonormal columns as well. In the end, randomized cluster basis $S_k^{(l)}$ have orthonormal columns: $\langle S_k^{(l)}(:, i), S_k^{(l)}(:, j) \rangle = \langle \tilde{S}_k^{(l)}(:, :) W(:, i), \tilde{S}_k^{(l)}(:, :) W(:, j) \rangle = \langle W(:, i), W(:, j) \rangle = \delta_{i,j}$.

We repeat this procedure for $T_k^{(l)}(\cdot, \tilde{\gamma}_k^{(l)})$ to obtain an orthonormal randomized basis in the same way.

Another improvement is to construct the random coefficient W using a tensor-network parameterization. Specifically, for each column $W(:, i)$, we model it either as a rank-one tensor product or as a tensor-train representation with rank $\tilde{r}$. This structured construction can further reduce computational cost and improve overall efficiency.

4.2.4 Algorithm and Complexity

In this section, we summarize the steps for obtaining $\{C_k^{(l)}\}_{k,l}$, the tensor-network representation of a density p, in Algorithm 9.

Algorithm 9 Algorithm for hierarchical tensor-network estimation (algorithm denoted as $\mathcal{A}(\cdot)$).

Require: Input $p : [n_1] \times \cdots \times [n_d] \to \mathbb{R}$. Sets of sketch functions $\{S_k^{(l)}\}_{k,l}$ and $\{T_k^{(l)}\}_{k,l}$. Rank at each level $\{r^{(l)}\}_l$.

1: **for** $l = 1, \cdots, L$ **do**
2: **for** $k = 1, \cdots, 2^{l-1}$ **do**
3: Obtain $p_i^{(l)}$ in (4.5) with $T_i^{(l)}$ and p, $i = 2k-1, 2k$.
4: Obtain $A_i^{(l)}$ in (4.7) with $S_i^{(l)}$ and $p_i^{(l)}$, $i = 2k-1, 2k$.
5: Obtain $B_k^{(l-1)}$ in (4.8) with $S_{2k-1}^{(l)}$, $S_{2k}^{(l)}$ and $p_k^{(l-1)}$.
6: Trim $A_{2k-1}^{(l)}$ and $A_{2k}^{(l)}$ with rank $r^{(l)}$ and solve for $G_k^{(l-1)}$ in (4.11) in terms of $C_k^{(l-1)}$ in (4.12).
7: **end for**
8: **end for**
9: Obtain $C_k^{(L)}$ in (4.13) for $k = 1, \cdots, 2^L$.
10: **Return** d-dimensional function represented in tensor-network $\{\{C_k^{(l)}\}_{k=1}^{2^l}\}_{l=0}^{L}$ as in Figure 4.5(B).

In practice, we apply this algorithm to empirical distribution $\hat{p}$, which is constructed from N independent and identically distributed samples. So $\hat{p}$ has at most N non-zero entries. We denote $\tilde{r}_{\texttt{max}} = \max_l \tilde{r}^{(l)}$ and $r_{\texttt{max}} = \max_l r^{(l)}$ for convenience where $\tilde{r}_{\texttt{max}} > r_{\texttt{max}}$. For

convenience, we assume cluster sketching order $t = 1$ in Section 4.2.3. To calculate the complexity, we assume that evaluating a function on a single entry is $O(1)$. For each specific k, l, $(k \in [2^l], l \in [L])$

1. $S_i^{(l)}, T_i^{(l)}$ can be computed in $O(\tilde{r}_{\texttt{max}} dN)$ time, $i = 2k-1, 2k$.
2. $p_i^{(l)}$ can be computed in $O(\tilde{r}_{\texttt{max}} N)$ time, $i = 2k-1, 2k$.
3. $A_i^{(l)}$ can be computed in $O(\tilde{r}_{\texttt{max}}^2 N)$ time, $i = 2k-1, 2k$.
4. $B_k^{(l-1)}$ can be computed in $O(\tilde{r}_{\texttt{max}}^3 N)$ time.
5. $\mathcal{P}_{r^{(l)}}(A_i^{(l)})$ can be computed in $O(\tilde{r}_{\texttt{max}}^2 r_{\texttt{max}})$ time, $i = 2k-1, 2k$.
6. $G_k^{(l-1)}$ can be computed in $O(\tilde{r}_{\texttt{max}}^3 r_{\texttt{max}} + \tilde{r}_{\texttt{max}}^2 r_{\texttt{max}}^2)$ time.
7. $C_k^{(l-1)}$ can be computed in $O(\tilde{r}_{\texttt{max}}^3 r_{\texttt{max}} + \tilde{r}_{\texttt{max}}^2 r_{\texttt{max}}^2 + \tilde{r}_{\texttt{max}} r_{\texttt{max}}^3)$ time.

The number of k, l is at most $d \log_2(d)$ in total. Note that the computation of sketching function in the first step can be implemented in a parallel way. Therefore, the total complexity of this algorithm is

$$O\left(d \log(d)\ (\tilde{r}_{\texttt{max}}^3 N + \tilde{r}_{\texttt{max}}^3 r_{\texttt{max}})\right)$$

which depends linearly on N and near linearly on d.

4.3 Theoretical Results

In this section, we provide analysis regarding the estimation error. Denote p^*, $\hat{p}$, $\tilde{p} = \mathcal{A}(\hat{p})$ as the ground truth distribution, empirical distribution in (4.1) and the tensor-network obtained from Algorithm 9. Recall that the total error is given by

$$p^* - \mathcal{A}(\hat{p}) = p^* - \mathcal{A}(p^*) + \mathcal{A}(p^*) - \mathcal{A}(\hat{p}), \tag{4.19}$$

where $\mathcal{A}$ is the proposed Algorithm 9.

We apply the following blanket assumption throughout the section.

Assumption 6. *We assume the ground truth distribution p^* could be represented by a hierarchical tensor-network with rank $\{r^{(l)}\}_{l=1}^{L}$ and further, the assumption of Theorem 6 is satisfied by $p = p^*$ with our choice of sketch functions.*

In this case, no approximation error is committed when applying Algorithm 9 to p^*, i.e. $\mathcal{A}(p^*) = p^*$. In practice, this may not be true, and we discuss such an error in Section 4.4 via numerical experiments. With such an assumption, the total error could be simplified as $p^* - \mathcal{A}(\hat{p}) = p^* - \tilde{p}$, which is the estimation error caused by the sample variance in $\hat{p}$. Our goal is to prove the stability of $\|p^* - \tilde{p}\|_F$ in terms of N and d.

We summarize the notations in this section:

1. We denote $G_k^{(l)} \to G_k^{(l)*}$ if we input p^* to $\mathcal{A}(\cdot)$, and $G_k^{(l)} \to \hat{G}_k^{(l)}$ if we input $\hat{p}$ to $\mathcal{A}(\cdot)$ where tensor core $G_k^{(l)}$ is defined in (4.11).

2. We let $A_k^{(l)} \to A_k^{(l)*}$ if we input p^* to $\mathcal{A}(\cdot)$, and $A_k^{(l)} \to \hat{A}_k^{(l)}$ if we input $\hat{p}$ to $\mathcal{A}(\cdot)$ where $A_k^{(l)}$ is defined in (4.7).

3. We let $B_k^{(l)} \to B_k^{(l)*}$ if we input p^* to $\mathcal{A}(\cdot)$, and $B_k^{(l)} \to \hat{B}_k^{(l)}$ if we input $\hat{p}$ to $\mathcal{A}(\cdot)$ where $B_k^{(l)}$ is defined in (4.8).

4. We let $p_k^{(l)} \to p_k^{(l)*}$ when $p \to p^*$ where $p_k^{(l)}$ is defined in (4.5). This encodes the ranges of various unfolding matrices of p^*. We also denote $\tilde{p}_k^{(l)}$ as estimated density at k-th node in l-th level, following the recursion level by level in (4.20):

$$\begin{aligned} \tilde{p}_k^{(L)}(\mathcal{I}_k^{(L)}, :) &= \hat{p}(\mathcal{I}_k^{(L)}, :) T_k^{(L)}, k \in [2^L], \\ \tilde{p}_k^{(l-1)} &= \tilde{p}_{2k-1}^{(l)} \hat{G}_k^{(l-1)} \tilde{p}_{2k}^{(l)T}, k \in [2^{l-1}], l \in [L]. \end{aligned} \tag{4.20}$$

In the last level, $\tilde{p}_k^{(L)}(\mathcal{I}_k^{(L)},:)$ reflects the sketched range of the empirical distribution $\hat{p}(\mathcal{I}_k^{(L)},\mathcal{J}_k^{(L)})$, a good candidate of marginal density to recover the distribution based on this hierarchical structure.

5. We use $\mathcal{P}_{r^{(l)}}(A_k^{(l)*})$ and $\mathcal{P}_{r^{(l)}}(\hat{A}_k^{(l)})$ to represent the best rank-$r^{(l)}$ approximation matrix of $A_k^{(l)*} \in \mathbb{R}^{\tilde{r}^{(l)}\times\tilde{r}^{(l)}}$ and $\hat{A}_k^{(l)} \in \mathbb{R}^{\tilde{r}^{(l)}\times\tilde{r}^{(l)}}$, respectively. We recall notation $\tilde{r}_{\texttt{max}} = \max_l \tilde{r}^{(l)}$.

6. We use ϵ_A to represent the maximum of the following Frobenius norm:$\epsilon_A = \max_{k,l} \|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F$.

7. We use $c_{A^\dagger}, c_{\hat{A}^\dagger}, c_S, c_{S^\dagger}, c_p, c_{\tilde{p}}$ as the upper bounds of the following terms:

$$\begin{aligned}&\|\left(A_k^{(l)*}\right)^\dagger\|_2 \le c_{A^\dagger}, \|\left(\mathcal{P}_{r^{(l)}}(\hat{A}_k^{(l)})\right)^\dagger\|_2 \le c_{\hat{A}^\dagger},\\ &\|S_k^{(l)}\|_2 \le c_S, \|\left(S_k^{(l)}\right)^\dagger\|_2 \le c_{S^\dagger}, \|p_k^{(l)*}\|_F \le c_p, \|\tilde{p}_k^{(l)}\|_F \le c_{\tilde{p}}, k \in [2^l], l \in [L]. \quad (4.21)\end{aligned}$$

We also use c_p as an upper bound in the following: $c_p \ge \|p_1^{(0)*}\|_F = \|p^*\|_F$.

Here we summarize our main results in the following theorem:

Theorem 7. *Suppose Assumption 5 holds and sketches are orthogonal, i.e. for each k, l, $\langle S_k^{(l)}(:,i), S_k^{(l)}(:,j)\rangle = \delta_{ij}$, $\langle T_k^{(l)}(:,i), T_k^{(l)}(:,j)\rangle = \delta_{ij}$, $i,j \in [\tilde{r}^{(l)}]$. Then for any $0 < \delta < 1$, when $c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 > 1$, the following inequality holds with probability at least $1-\delta$:*

$$\|\tilde{p} - p^*\|_F \le \tilde{C}(c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^{\log_2 d} \frac{\log(\frac{2\tilde{r}_{\texttt{max}} d \log_2 d}{\delta})}{\sqrt{N}} \quad (4.22)$$

where $\tilde{C} = 3\sqrt{\tilde{r}_{\texttt{max}}}\left(1 + \frac{c_{\tilde{p}}^2(c_{\hat{A}^\dagger}^2 + 6c_{A^\dagger}c_{\hat{A}^\dagger}^2 c_p + 6c_{A^\dagger}^2 c_{\hat{A}^\dagger} c_p)}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1}\right)$.

Proof. By assembling Lemma51 and Lemma52 in C.2, we have

$$\|\tilde{p} - p^*\|_F \leq \kappa_{p^{(0)}} \epsilon_A = (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^{\log_2 d} \left(1 + \frac{c_{\tilde{p}}^2 \kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1}\right) \epsilon_A, \tag{4.23}$$

where $\kappa_G = c_{\hat{A}^\dagger}^2 + 6c_{A^\dagger} c_{\hat{A}^\dagger}^2 c_p + 6c_{A^\dagger}^2 c_{\hat{A}^\dagger} c_p$. By plugging upper bound ϵ_A from Lemma 53 into (4.23), we have the following inequality holds with probability at least $1 - \delta$:

$$\|\tilde{p} - p^*\|_F \leq \tilde{C}(c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^{\log_2 d} \frac{\log(\frac{2\tilde{r}_{\mathtt{max}} d \log_2 d}{\delta})}{\sqrt{N}}, \tag{4.24}$$

where $\tilde{C} = 3\sqrt{\tilde{r}_{\mathtt{max}}} \left(1 + \frac{c_{\tilde{p}}^2 (c_{\hat{A}^\dagger}^2 + 6c_{A^\dagger} c_{\hat{A}^\dagger}^2 c_p + 6c_{A^\dagger}^2 c_{\hat{A}^\dagger} c_p)}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1}\right)$. □

Inspired by [179], the theorem is aimed to bound the error in terms of the Frobenius norm. In the following parts, we would provide two remarks about this theorem. Next we show some preliminaries in C.1 and present the components including Lemma 51, Lemma 52, and Lemma 53 in C.2.

Remark 2. *We remark that the assumption $c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 > 1$ can always be satisfied by a sufficient large choice of the constant $c_{A^\dagger}$.*

Remark 3. *Furthermore, based on the fact that p^* should be density, we could have bounds for c_p to simplify the result. More specifically, the sum of the absolute value of entries of p^* are 1 since p^* is a density. Thus $\|p_k^{(l)*}\|_F$ has the following upper bound: $\|p_k^{(l)*}\|_F \leq \|p^* T_k^{(l)}\|_F \leq \|p^*\|_F \|T_k^{(l)}\|_2 \leq 1$ for every k, l and therefore $c_p \leq 1$. Besides, if $c_{\tilde{p}} \leq 1$ also holds, the simplified inequality holds with probability at least $1 - \delta$:*

$$\|\tilde{p} - p^*\|_F \leq \tilde{C}' d (c_{A^\dagger})^{2\log_2 d} \frac{\log(\frac{2\tilde{r}_{max} d \log_2 d}{\delta})}{\sqrt{N}}, \tag{4.25}$$

where $\tilde{C}' = 3\sqrt{\tilde{r}_{max}} \left(1 + \frac{c_{\hat{A}^\dagger}^2 + 6c_{A^\dagger} c_{\hat{A}^\dagger}^2 + 6c_{A^\dagger}^2 c_{\hat{A}^\dagger}}{2c_{A^\dagger}^2 - 1}\right)$.

If $c_p \leq 1 \leq c_{\tilde{p}}$, then the following inequality holds with probability at least $1-\delta$:

$$\|\tilde{p} - p^*\|_F \leq \tilde{C}'' d (c_{A^\dagger} c_{\tilde{p}})^{2\log_2 d} \frac{\log(\frac{2\tilde{r}_{max} d \log_2 d}{\delta})}{\sqrt{N}}, \tag{4.26}$$

where $\tilde{C}'' = 3\sqrt{\tilde{r}_{max}} \left(1 + \frac{c_{\tilde{p}}^2 (c_{\hat{A}^\dagger}^2 + 6 c_{A^\dagger} c_{\hat{A}^\dagger}^2 + 6 c_{A^\dagger}^2 c_{\hat{A}^\dagger})}{2 c_{A^\dagger}^2 c_{\tilde{p}}^2 - 1}\right)$.

4.4 Numerical Results

In this section, we demonstrate the success of the algorithm on various one-dimensional and two-dimensional Ising models. [1] We recall notation p^*, $\hat{p}$, $\tilde{p} = \mathcal{A}(\hat{p})$ as the ground-truth distribution, the empirical distribution and the tensor-network represented distribution, respectively. We can compute the relative error with respect to p^*:

$$\epsilon_p = \frac{\|\tilde{p} - p^*\|_F}{\|p^*\|_F}$$

For the choice of sketch function in Section 4.2.3, we are concerned with Ising-like models where $n = 2$.

4.4.1 One-dimensional Ising Model

Ferromagnetic Ising model

We consider the following generalization of the one-dimensional Ising model. Define $p^* : \{-1, 1\}^d \to \mathbb{R}$ by

$$p^*(x_1, \cdots, x_d) \propto \exp(-\beta \sum_{i,j=1}^{d} J_{ij} x_i x_j)$$

1. The implementation of hierarchical tensor sketching can be found at https://github.com/IvanPeng0414/Hierarchical-Tensor-Sketching.

where $\beta > 0$ reflects the inverse of temperature in this model and the interaction coefficient J_{ij} is given by

$$J_{ij} = \begin{cases} -1/2, & |i-j| = 1 \\ -1/6, & |i-j| = 2 \\ 0, & \texttt{otherwise} \end{cases}$$

giving interactions between both nearest and next-nearest neighbors. We first investigate the error with respect to the number of samples (N) and show the resulting error in Figure 4.6 for various temperatures. A typical distribution is visualized in Figure 4.9(A). The distribution concentrates on two states, where one of them has entirely positive signs and the other has all negative signs.

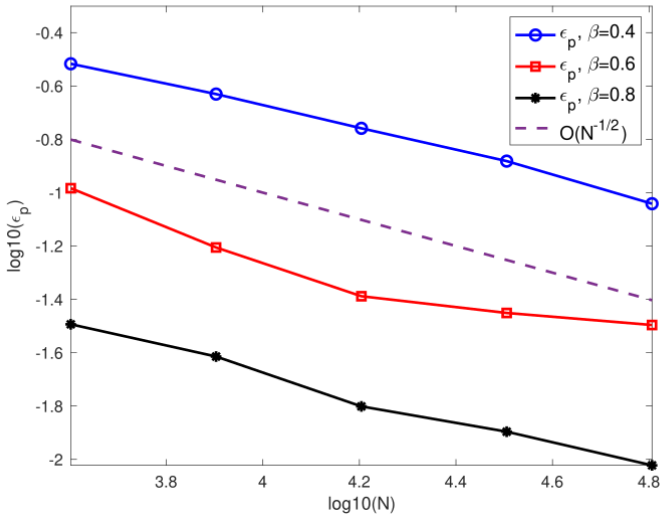


Figure 4.6: Error for next neighbor one-dimensional Ising model with $d = 16$. Blue, red, and black curves represent cases with $\beta = 0.4$, $\beta = 0.6$, and $\beta = 0.8$, respectively. Dashed curve reflects a reference curve $O(N^{-1/2})$.

Based on the results obtained, it can be observed that the error decreases as $\frac{1}{\sqrt{N}}$, which satisfies the Monte-Carlo rate. Additionally, it is noticeable that the error is smaller in the case of lower temperature examples. This can be explained by the fact that the distribution is more concentrated at lower temperatures, giving less variance. Additionally, we plot the

error v.s. d in Figure 4.7 with temperature $\beta = 0.6$ and $N = 64000$ empirical samples and observe a mild growth with the dimensionality d.

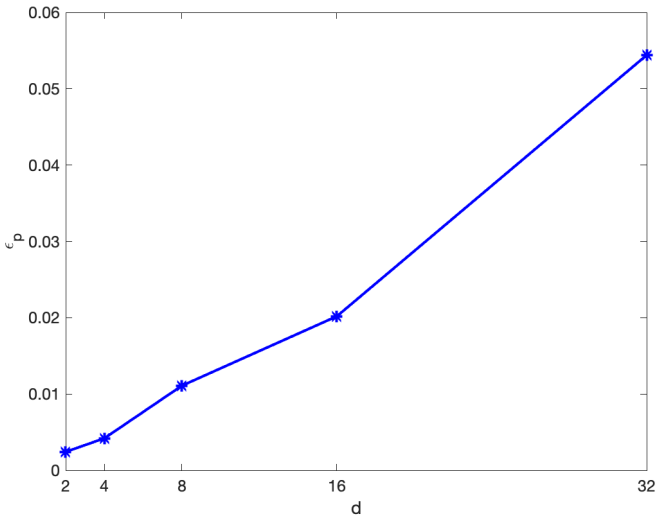


Figure 4.7: Change of error with respect to d in one-dimensional ferromagnetic Ising model with $\beta = 0.6$.

Antiferromagnetic Ising model

Another important case is the antiferromagnetic example, where the ground-truth distribution has the same form

$$p^*(x_1, \cdots, x_d) \propto \exp(-\beta \sum_{i,j=1}^{d} J_{ij} x_i x_j)$$

but the interaction coefficients J_{ij} are positive valued:

$$J_{ij} = \begin{cases} 1/2, & |i-j| = 1 \\ 1/6, & |i-j| = 2 \\ 0, & \texttt{otherwise} \end{cases}$$

Figure 4.8 showcases the results obtained by hierarchical tensor network. Again, the error decreases in terms of N at a rate of $1/\sqrt{N}$. However, when compared to the ferromagnetic Ising model, the error is larger. To gain a better understanding of this difference, we could visualize the empirical distribution. In Figure 4.9, it is evident from the antiferromagnetic type distribution that it is less concentrated and has more small local peaks compared to the ferromagnetic example at the same temperature, thus more samples and more detailed sketches are required to accurately recover the antiferromagnetic case. Additionally, the distribution of antiferromagnetic Ising model concentrates on two states, each of which have mixed $+1$ and -1 variables and are visualized in Figure 4.9. Furthermore, states around the two main states occur with non-negligible probability. This could explain why the error of antiferromagnetic Ising model would be larger than the one in ferromagnetic Ising model.

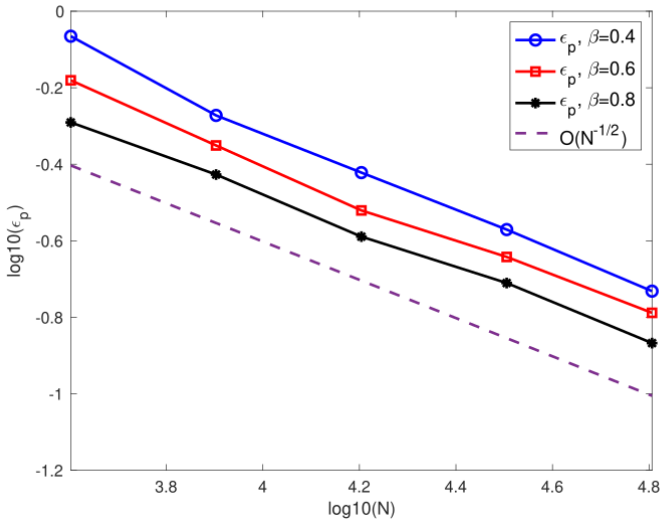
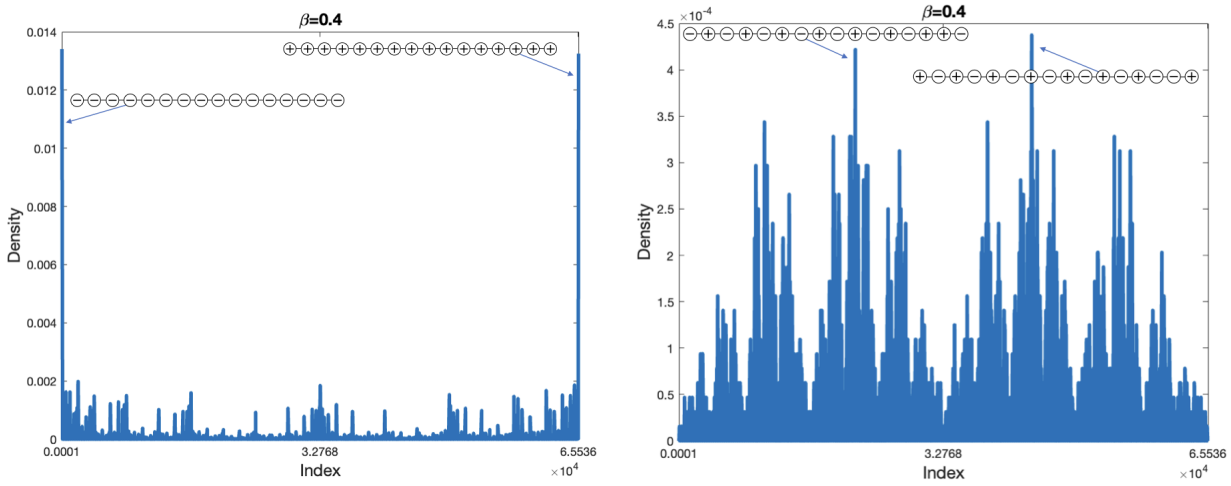


Figure 4.8: Error for next neighbor antiferromagnetic Ising model with $d = 16$. Blue, red, and black curves represent cases with $\beta = 0.4$, $\beta = 0.6$, and $\beta = 0.8$, respectively. Dashed curve reflects a reference curve $O(N^{-1/2})$.



(a) Ferromagnetic Ising model.

(b) Antiferromagnetic Ising model.

Figure 4.9: The empirical distribution with $N = 64000$, $d = 16$ and $\beta = 0.4$ of ferromagnetic Ising model (Left) and antiferromagnetic Ising model (Right). Four representative states are included in the figure. "+" and "−" correspond to $x_i = +1$ and $x_i = -1$, respectively.

4.4.2 Two-dimensional Ising Model

The next example is two-dimensional Ising model. Here we only consider the interaction between nearest neighbor. The probability distribution is given by the following:

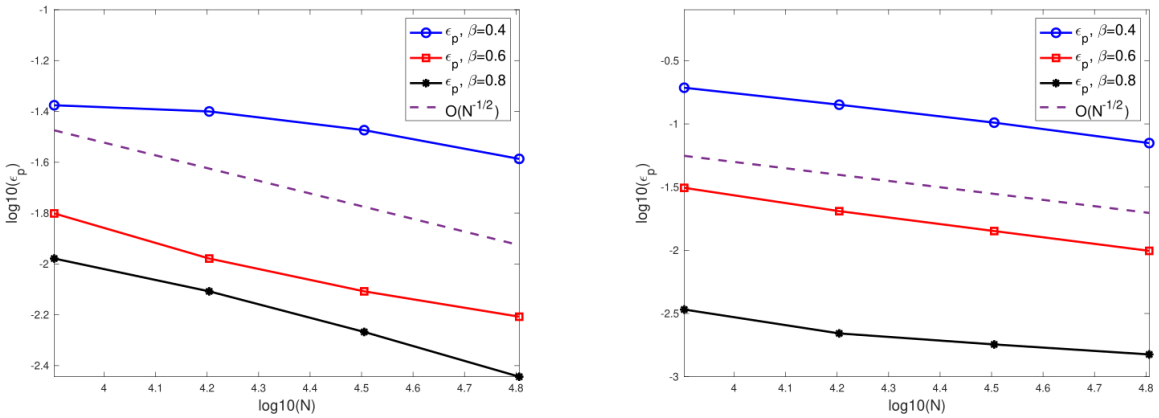
$$p^*(\{x_{i,j}\}_{i,j=1,\cdots,d}) \propto \exp\left(-\beta \sum_{i=1}^{d}\sum_{j=1}^{d}(Jx_{i,j}x_{i+1,j} + Jx_{i,j}x_{i,j+1})\right)$$

where $\beta > 0$. We consider periodic boundary condition in the problem, which means $x_{i,d+1} = x_{i,1}$ and $x_{d+1,j} = x_{1,j}$ for $i, j = 1, \cdots, d$.

Ferromagnetic Ising model

For the ferromagnetic Ising model we take $J = -1$. The left figure in Figure 4.10 displays results for different numbers of samples at different temperatures, for a 4×4 Ising model, while the right figure gives the results for an 8×8 system. Similar to the one-dimensional Ising model, the error decreases in terms of N at a rate of $1/\sqrt{N}$. For two-dimensional

ferromagnetic Ising model, we visualize five typical samples in example $\beta = 0.4$ of 4×4 domain. In Figure 4.11, we find that samples from the distribution would concentrate on two states: one state has all positive signs and the other has all negative signs.



(a) 4×4 domain. (b) 8×8 domain.

Figure 4.10: Error of nearest neighbor Ising model in several temperature scenarios. Blue, red, and black curves represent cases with $\beta = 0.4$, $\beta = 0.6$, and $\beta = 0.8$, respectively. Dashed curve reflects a reference curve $O(N^{-1/2})$. Left: 4×4 domain; Right: 8×8 domain.

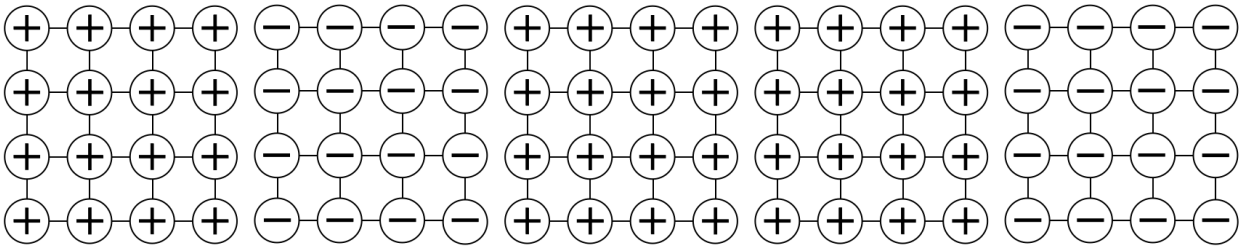


Figure 4.11: Visualization of five samples in 4×4 ferromagnetic Ising model with $\beta = 0.4$. Here "+" represents $x_{i,j} = +1$ and "−" represents $x_{i,j} = -1$ for $i, j = 1, \cdots, 4$.

Antiferromagnetic Ising model

In this section, we study the antiferromagnetic two-dimensional Ising model. The probability distribution can be expressed as:

$$p^*(\{x_{i,j}\}_{i,j=1,\cdots,d}) \propto \exp\left(-\beta \sum_{i=1}^{d}\sum_{j=1}^{d}(Jx_{i,j}x_{i+1,j} + Jx_{i,j}x_{i,j+1})\right)$$

with $J = 1$. In Figure 4.12, five typical samples from this type of distribution are demonstrated, which shows oscillatory sign pattern. This stands in contrast to the ferromagnetic Ising model. Figure 4.13 presents the results obtained in both the 4×4 and 8×8 models. The algorithm exhibits decent accuracy with a sufficient number of samples.

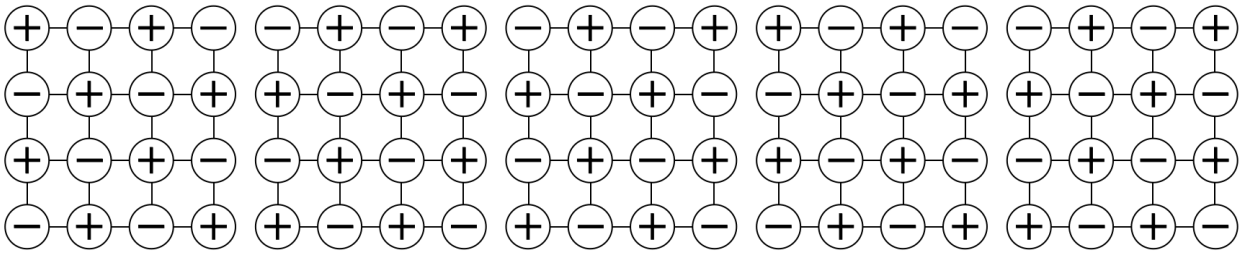
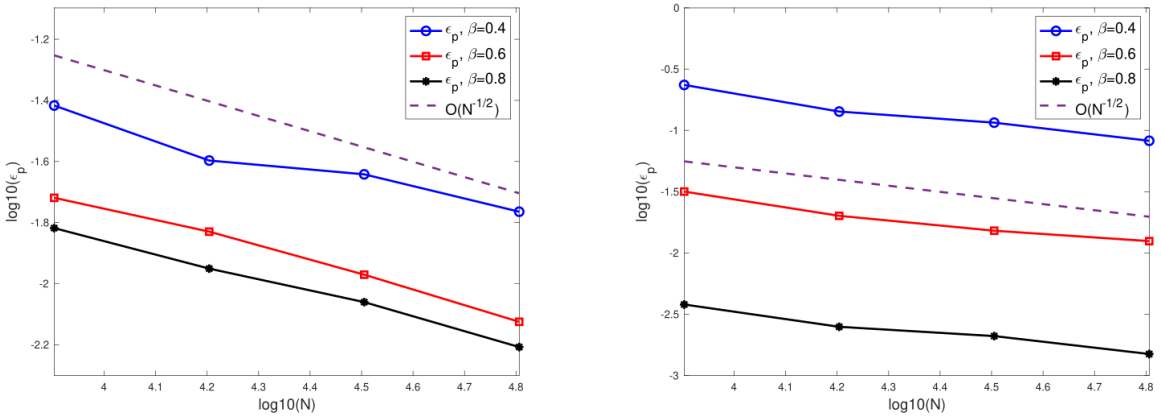


Figure 4.12: Visualization of five samples in 4×4 antiferromagnetic Ising model with $\beta = 0.4$. Here "+" represents $x_{i,j} = +1$ and "−" represents $x_{i,j} = -1$ for $i, j = 1, \cdots, 4$.

Choice of rank

The choice of rank in this algorithm is crucial as it directly affects the model's complexity, i.e., the number of parameters in the model. Like many machine learning methods, there is a trade-off when selecting the rank, the hyperparameters in this model: If the rank is too small, the model may not be able to fit the distribution well due to a lack of representation power. On the other hand, if the rank is too large, it may lead to overfitting which results in a large estimation error.

To illustrate the impact of rank, we consider a 4×4 ferromagnetic Ising model with



(a) 4×4 domain. (b) 8×8 domain.

Figure 4.13: Error of nearest neighbor antiferromagnetic Ising model with several temperatures. Blue, red, and black curves represent cases with $\beta = 0.4$, $\beta = 0.6$, and $\beta = 0.8$, respectively. Dashed curve reflects a reference curve $O(N^{-1/2})$. Left: 4×4 domain; Right: 8×8 domain.

$\beta = 0.2$. Our first experiment is to study the effect of rank at the top level $r_1^{(0)}$ (rank of $G_1^{(0)}$) on the estimation error with a fixed number of samples N. We compare the error $\epsilon_p = \|\tilde{p} - p^*\|_F / \|p^*\|_F$ with the relative approximation error $\epsilon_{\texttt{approx}} = \|\mathcal{A}(p^*) - p^*\|_F / \|p^*\|_F$, i.e. the error when we replace $\hat{p}$ with the ground truth p^* when applying $\mathcal{A}(\cdot)$. In Figure 4.14, we show that the best rank that achieves the smallest error ϵ_p for each N, increases as N increases. When N is large, the error of $\mathcal{A}(\hat{p})$ is converging to the approximation error $\epsilon_{\texttt{approx}}$. We can explain this trend as follows: when N is small, by introducing a bias error via using a smaller rank, we can reduce the variance in estimation. In contrast, with large number of samples, we can afford to use a large rank to reduce the approximation error.

Our second experiment analyzes how the error changes with respect to N for fixed ranks at the top level. Using the same example as the first test, we depict the change in the total error ϵ_p with respect to the number of samples for two choices of rank, 6 and 10, respectively, in Figure 4.15. It is apparent that the relative approximation error is smaller with rank = 10 than with rank = 6. However, when we compute the estimation error $(\epsilon_p - \epsilon_{\texttt{approx}})$, the case of rank = 6 shows faster convergence. We can explain this observation using the bias-

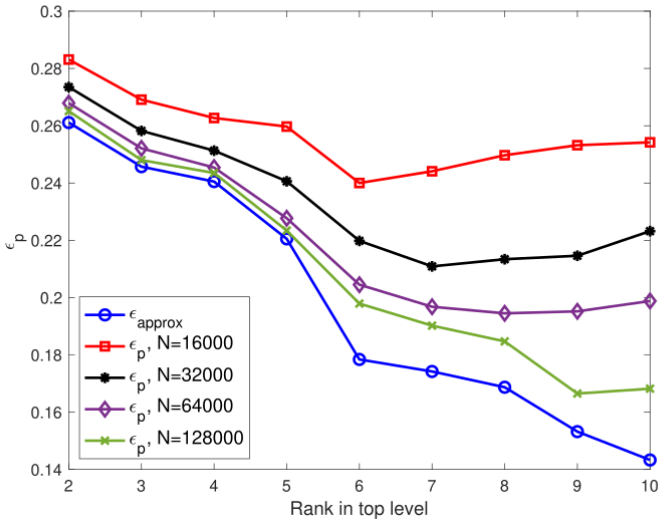


Figure 4.14: The influence of truncated rank in top level on error in 4×4 ferromagnetic Ising model with $\beta = 0.2$. The approximation error $\epsilon_{\texttt{approx}}$ is shown in the line with circle marker; errors of empirical distribution with $N = 16000$, $N = 32000$, $N = 64000$ and $N = 128000$ are shown in line with square, star, diamond and cross marker, respectively.

variance trade-off. The bias error is reflected by $\epsilon_{\texttt{approx}}$, which depends solely on the model complexity and problem difficulty. In this figure, although rank $= 10$ provides smaller bias error, the total error is larger than the rank $= 6$ case when number of samples is small. This again shows that with limited number of samples, one might want to commit a bias error to reduce the variance in the estimator $\tilde{p} = \mathcal{A}(\hat{p})$.

4.5 Conclusion

We present an optimization-free method for constructing a hierarchical tensor-network that approximates high-dimensional probability density using empirical distributions. Our approach involves sketching techniques from randomized linear algebra to form the tensor-network, with a computational complexity of $O(Nd \log d)$. We further demonstrate the success of the algorithm in several numerical examples concerning Ising-like models.

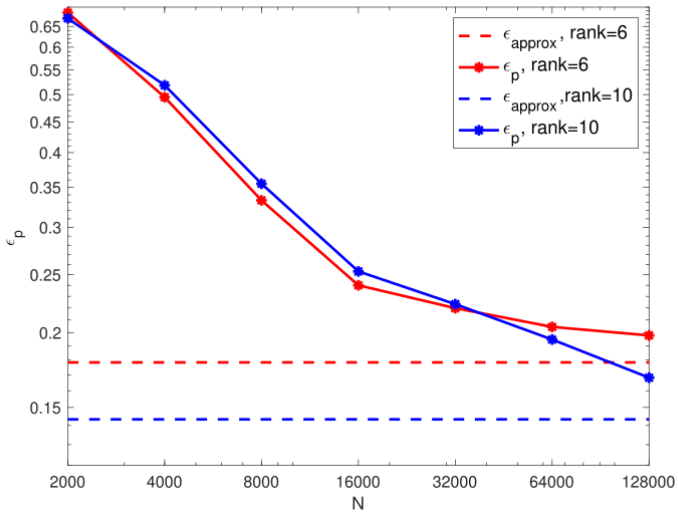


Figure 4.15: Change of error with respect to number of samples for two choices of rank in top level in 4×4 ferromagnetic Ising model with $\beta = 0.2$. The benchmark line given by relative approximation error is shown as red and blue dashed line for rank 6 and 10, respectively. The convergence curves are shown as red and blue star lines for rank 6 and 10, respectively.

CHAPTER 5

OPTIMIZATION-FREE DIFFUSION MODEL

5.1 Introduction

Generative modeling, one of most active areas in machine learning, aims to learn the underlying data distribution from samples so that new samples can be generated that closely resemble the original dataset. Among the leading approaches, diffusion models [104] have emerged as a powerful framework. These models rely on estimating the score function—the gradient of the log-probability density—to model and generate samples from complex distributions. More specifically, diffusion models typically learn the score function $s(t, x)$ by minimizing the following cost functional:

$$\min_{s(t,x)} \int_0^T \int_{\mathbb{R}^d} \frac{1}{2} \|s(t,x) - \nabla \log \rho_t(x)\|^2 \rho_t(x) \mathrm{d}x \mathrm{d}t. \tag{5.1}$$

Here, T is a chosen time horizon and $\rho_t(x)$ is the law of x_t governed by the overdamped Langevin dynamics:

$$dx_t = -\nabla V(x_t)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}w_t, \quad x_0 \sim \rho_0, \tag{5.2}$$

where V is a user-specified potential function, β is the inverse temperature, and w_t is standard Brownian motion. This dynamic gradually transforms initial samples from the data distribution ρ_0 toward a tractable base distribution ρ_∞, which corresponds to the stationary distribution of the process:

$$\rho_\infty(x) = \frac{1}{Z} \exp(-\beta V(x)), \tag{5.3}$$

where Z is the normalization constant.

Once the score function is learned, new samples can be generated by simulating the

reverse-time stochastic differential equation (SDE) corresponding to (5.2), starting from samples generated from the base distribution ρ_∞ (which approximates ρ_T when T is sufficiently large) [9] and then propagating backwards in time:

$$d\tilde{x}_{T-t} = \Big(\nabla V(\tilde{x}_{T-t}) + 2\beta^{-1} s(T-t, \tilde{x}_{T-t})\Big)\,\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}w_{T-t}, \tilde{x}_T \sim \rho_\infty. \tag{5.4}$$

After evolving this process from $t = T$ to $t = 0$, the resulting samples $\tilde{x}_0$ are the newly generated data that closely resemble the original dataset.

To represent the score function $s(t, x)$ one usually employs neural networks, achieving state-of-the-art performance in generating high-quality images, audio, and other complex data types [203, 44, 204, 205, 152]. In terms of theoretical guarantees, several recent works [128, 129] have shown that a small L_2 error of the estimated score function implies a small discrepancy between the distribution generated by the reverse-time SDE (5.4) and the underlying ground truth distribution. In the neural network setting, it is often difficult to rigorously justify the assumption of a small score estimation error due to the challenges associated with non-convex optimization and the lack of guarantees on global convergence. In this work, we address this issue by exploring an alternative approach aside of neural networks, aiming to understand and quantify how accurately the score function can be estimated based on the number of degrees of freedom in the perturbative regime.

Contributions

Here we summarize our contributions:

1. We propose an optimization-free and forward SDE-free approach for diffusion models. Specifically, when representing the score function by eigenfunctions of the backward Kolmogorov operator associated with base distribution (5.3), the score can be obtained via solving linear systems. Furthermore, the coefficient matrix in the linear systems

can be computed without relying on any forward SDE simulation.

2. We provide a theoretical analysis of the score estimation error using perturbation theory, under the assumption that the underlying density is a perturbation of the base distribution (5.3). We show that the score error can be bounded in terms of perturbation scale parameter and the number of basis functions used to represent the score.

3. We overcome the curse-of-dimensionality by projecting the score function onto the span of a *cluster basis*, which is a sparse subset of the high-dimensional eigenfunctions of the backward Kolmogorov operator. This basis captures low-order correlations between the variables. The coefficients for the score function can be obtained using fast numerical techniques for solving linear system in high-dimensional settings. As a result, the overall computation cost is reduced to $O(Nd^2 + N_t d^2)$ cost where N is the number of initial samples and N_t is the number of time steps. We demonstrate the effectiveness of our method on several high-dimensional Boltzmann distributions and real-world image datasets, achieving favorable results.

Related Work

Diffusion models overview: Research on diffusion models has primarily centered around two dominant formulations. The first approach is based on forward SDE simulation [92, 200, 205, 202], where a stochastic differential equation (typically an Ornstein–Uhlenbeck process) is used to gradually diffuse data into noise. The score function is then estimated by training a neural network. For example, the Denoising Diffusion Probabilistic Model (DDPM) [92, 200] uses a pair of Markov chains to progressively corrupt and subsequently recover the data, with the reverse process learned via deep neural networks. Score-based generative models (SGMs) [203, 204, 219] apply a sequence of increasingly strong Gaussian noise to the data and learn the corresponding score functions across time. The second

formulation is based on flow matching [134, 140, 138], which directly learns a time-dependent vector field that couples the target and base distributions, rather than estimating score functions through a forward stochastic process. Extensions of this approach include method based on stochastic interpolants [7, 5, 6], which interpolate between distributions using latent variables. For an overview of these methods, see [238, 28, 145]. Our work falls into the first category, where the forward SDE is modeled as an overdamped Langevin dynamics with a user-specified potential function. Importantly, the stationary distribution associated with this dynamic is not restricted to the Gaussian case; instead, we generalize it to a broader class of distributions that remain amenable to efficient sampling.

Other related works have also explored training-free diffusion models, but their underlying philosophy differs substantially from that of the present work. Some of these methods focus on conditional diffusion models and investigate how to leverage a pre-trained unconditional diffusion model to achieve strong conditional generation performance without any additional training. For example, [241] introduces a flexible and adaptive energy function to characterize the conditional density. The work [240] provides a general framework for training-free guidance in the reverse sampling stage, including several existing approaches such as [15], with the goal of improving sampling efficiency. In contrast, our work considers a different representation of score function. Rather than relying on a pre-trained diffusion model, we focus on reducing the computational cost of score estimation in a training-free setting without introducing any pre-trained network.

Error analysis of score estimation: Many recent works have provided an error analysis for score estimation in diffusion models when using neural network as the model class. In terms of approximation error, [31] considers a local Taylor approximation of the score function using neural networks, focusing on data supported on low-dimensional linear subspaces. The work demonstrates that a simplified U-Net architecture with linear encoder and decoder can achieve an efficient score approximation. In a different direction, [162] develops the theo-

retical foundations for score approximation when the initial density lies in a Besov space. In terms of statistical error, [18] analyzes the Rademacher complexity of neural network classes for score representation and [31, 162] provide nonparametric statistical perspectives on this type of error. In terms of optimization error, [83] studies the optimization guarantees for score estimation using two-layer neural networks. Additional related works are surveyed in the comprehensive overview [32]. In contrast to these approaches, our work is completely based on linear regression, thereby avoiding the intractable optimization error often associated with complex neural network architectures. Furthermore, we leverage perturbation theory to provide a clear and rigorous analysis of the approximation error, providing a new perspective on score estimation.

We conclude this section by outlining the organization of the work. In Section 5.2, we introduce the main idea behind our method, explaining how the score function is represented using a set of eigenfunctions. Section 5.3 provides a detailed explanation of our approach for single-variable distributions and Section 5.4 extends the discussion to the high-dimensional case, where we incorporate additional techniques to surmount the curse of dimensionality. In Section 5.5, we present a theoretical analysis of the approximation error based on perturbation theory. Section 5.6 contains our numerical results, which include both synthetic examples from Boltzmann distributions and real-world data such as the MNIST dataset. Finally, Section 5.7 concludes the work.

5.2 Main Idea

In this work, we propose a new perspective on diffusion models that relies entirely on numerical linear algebra, rather than neural network representations, thereby yielding an optimization-free approach. Specifically, we represent the score function as an expansion in

a set of basis functions $\{f_k\}_{k\in\mathcal{S}}$ where $\mathcal{S}$ is a chosen index set:

$$s^{(i)}(t,x) = \sum_{k\in\mathcal{S}} c_k^{(i)}(t) f_k(x) - \beta\partial_{x_i} V(x) = (c^{(i)}(t))^T f(x) - \beta\partial_{x_i} V(x), \tag{5.5}$$

where $s^{(i)}(t,x)$ is i-th component of the score $s(t,x)$ and $c_k^{(i)}(t)$ are the corresponding time-dependent coefficients. Note that, according to (5.3), the ground-truth score function in the limit $t\to\infty$ is $s(\infty,x) = \nabla\log\rho_\infty(x) = -\beta\nabla V(x)$. Thus, the advantage of including the term $-\beta\nabla V(x)$ in (5.5) is that the corresponding coefficients $c_k^{(i)}(t)$ naturally decline to zero as $t\to\infty$, simplifying the score estimation for large t.

To estimate the score function, the objective in (5.1) can be equivalently reformulated using integration by parts [104] as:

$$\min_{s(t,x)} \int_0^T \int_{\mathbb{R}^d} \left(\frac{1}{2}\|s(t,x)\|^2 + \mathrm{div}(s(t,x))\right) \rho_t(x)\mathrm{d}x\mathrm{d}t, \tag{5.6}$$

which is often more convenient for optimization. Given a time grid $t_{\text{grid}} \subset [0,T]$, the minimization problem in (5.6) reduces to independently minimizing the coefficients $c^{(i)}(t)$ for each time $t\in t_{\text{grid}}$ and each dimension index i:

$$\begin{aligned}\min_{c^{(i)}(t)} \frac{1}{2}\sum_{k,k'\in\mathcal{S}} c_k^{(i)}(t)c_{k'}^{(i)}(t)\int \rho_t(x) f_k(x) f_{k'}(x)\mathrm{d}x \\ + \sum_{k\in\mathcal{S}} c_k^{(i)}(t)\int \rho_t(x)\left(\partial_{x_i} f_k(x) - \beta f_k(x)\partial_{x_i}V(x)\right)\mathrm{d}x.\end{aligned} \tag{5.7}$$

For simplicity, we denote by $A(t)\in\mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ and $b^{(i)}(t)\in\mathbb{R}^{|\mathcal{S}|\times 1}$ the matrices:

$$A_{k,k'}(t) = \int \rho_t(x) f_k(x) f_{k'}(x)\mathrm{d}x, \quad b_k^{(i)}(t) = \int \rho_t(x)\left(\partial_{x_i} f_k(x) - \beta f_k(x)\partial_{x_i}V(x)\right)\mathrm{d}x. \tag{5.8}$$

The optimization problem (5.7) becomes

$$\min_{c^{(i)}(t)\in\mathbb{R}^{|\mathcal{S}|\times 1}} \frac{1}{2} c^{(i)}(t)^T A(t) c^{(i)}(t) + c^{(i)}(t)^T b^{(i)}(t) \tag{5.9}$$

whose solution satisfies the linear system $A(t)c^{(i)}(t) = -b^{(i)}(t)$. To simplify the notation, we aggregate the solutions for all dimension indices into a single linear system:

$$A(t)C(t) = -B(t), \tag{5.10}$$

where $C(t) = [c^{(1)}(t), \cdots, c^{(d)}(t)] \in \mathbb{R}^{|\mathcal{S}|\times d}$ and $B(t) = [b^{(1)}(t), \cdots, b^{(d)}(t)] \in \mathbb{R}^{|\mathcal{S}|\times d}$. By repeating this procedure for all $t \in t_{\text{grid}}$ and each dimension index i, we complete the computation of the score function $s(t, x)$. We refer to our approach as *optimization-free* because solving (5.10) can be efficiently handled via standard linear system solvers, without the need for iterative optimization methods such as gradient descent.

The remaining task is to efficiently compute $A(t)$ and $b^{(i)}(t)$ in (5.8). A common approach is to estimate the integrals inside $A(t)$ and $b^{(i)}(t)$ via Monte Carlo integration, using samples x_t generated by the Euler-Maruyama scheme (or other numerical SDE methods) applied to (5.2). This method incurs a dominant computational cost of $O(NN_t|\mathcal{S}|^2)$, where N is the sample size and N_t is the number of time steps. In contrast, we propose an alternative approach that avoids generating x_t at each time t, allowing us to compute the required integrals more efficiently and significantly reduce the overall computational cost.

Here we provide the intuition of our approach. Note that the overdamped Langevin dynamics in (5.2) is associated with a backward Kolmogorov operator

$$\mathcal{L} = -\nabla V(x) \cdot \nabla + \beta^{-1}\Delta, \tag{5.11}$$

whose adjoint operator, known as the Fokker-Planck operator, describes the time evolution

of probability density $\rho_t(x)$:

$$\partial_t \rho_t(x) = \mathcal{L}^* \rho_t(x), \quad \text{where} \quad \mathcal{L}^* \rho_t(x) = \nabla \cdot (\nabla V(x) \rho_t(x)) + \beta^{-1} \Delta \rho_t(x). \tag{5.12}$$

Therefore, if we choose the basis functions $\{f_k\}_{k \in \mathcal{S}}$ as the eigenfunctions of the backward Kolmogorov operator $\mathcal{L}$:

$$\mathcal{L} f_k = \lambda_k f_k,$$

the integration of these eigenfunctions w.r.t. the time-evolving density $\rho_t(x)$ is straightforward:

$$\int f_k(x) \rho_t(x) \mathrm{d}x = \int f_k(x) e^{\mathcal{L}^* t} \rho_0(x) \mathrm{d}x = \int \rho_0(x) e^{\mathcal{L} t} f_k(x) \mathrm{d}x = e^{\lambda_k t} \int f_k(x) \rho_0(x) \mathrm{d}x, \tag{5.13}$$

where λ_k is the corresponding eigenvalue. Given data samples $\{x^{(j)}\}_{j=1}^N$ from the initial distribution $\rho_0(x)$, the integral $\int f_k(x) \rho_0(x) \, \mathrm{d}x$ in the right hand side of (5.13) can be approximated via Monte Carlo integration as $\frac{1}{N} \sum_{j=1}^N f_k(x^{(j)})$. This allows us to approximate the integrals involving $\rho_t(x)$ through (5.13), without simulating the forward SDE as is generally done. In order to efficiently exploit equation (5.13) in the computation of the integrals inside $A(t)$ and $b^{(i)}(t)$, we project the product $f_k(x) f_{k'}(x)$ and derivatives $\partial_{x_i} f_k(x) - \beta f_k(x) \partial_{x_i} V(x)$ onto the span of eigenfunctions $\{f_k\}_{k \in \mathcal{S}'}$, i.e.

$$f_k(x) f_{k'}(x) = \sum_{l \in \mathcal{S}'} u_l^{(k,k')} f_l(x), \quad \partial_{x_i} f_k(x) = \sum_{l \in \mathcal{S}'} v_l^{(i,k)} f_l(x), \quad \partial_{x_i} V(x) = \sum_{l \in \mathcal{S}'} q_l^{(i)} f_l(x). \tag{5.14}$$

Then we are able to compute $A(t)$ and $b^{(i)}(t)$ following (5.13):

$$A_{k,k'}(t) = \sum_{l \in \mathcal{S}'} u_l^{(k,k')} \int e^{\mathcal{L} t} (f_l(x)) \rho_0(x) \mathrm{d}x = \sum_{l \in \mathcal{S}'} u_l^{(k,k')} e^{\lambda_l t} \int f_l(x) \rho_0(x) \mathrm{d}x, \tag{5.15}$$

$$
\begin{aligned}
b_k^{(i)}(t) &= \int e^{\mathcal{L}t} \left(\sum_{l \in \mathcal{S}'} v_l^{(i,k)} f_l(x) - \beta \sum_{j \in \mathcal{S}'} q_j^{(i)} f_k(x) f_j(x) \right) \rho_0(x) \mathrm{d}x \\
&= \int e^{\mathcal{L}t} \left(\sum_{l \in \mathcal{S}'} v_l^{(i,k)} f_l(x) - \beta \sum_{j \in \mathcal{S}'} q_j^{(i)} \sum_{l \in \mathcal{S}'} u_l^{(k,j)} f_l(x) \right) \rho_0(x) \mathrm{d}x \\
&= \sum_{l \in \mathcal{S}'} v_l^{(i,k)} e^{\lambda_l t} \int f_l(x) \rho_0(x) \mathrm{d}x - \beta \sum_{l \in \mathcal{S}'} (\sum_{j \in \mathcal{S}'} q_j^{(i)} u_l^{(k,j)}) e^{\lambda_l t} \int f_l(x) \rho_0(x) \mathrm{d}x. \quad (5.16)
\end{aligned}
$$

The choice of the new set $\mathcal{S}'$ depends on the base distribution and its associated eigenfunctions, ensuring that the representation in (5.14) holds with low or even no interpolation error. Importantly, Monte Carlo integration is required only once using $\rho_0(x)$ at the initial step, and the resulting quantities can be reused across all time steps t. This strategy significantly reduces computational complexity compared to conventional simulation-based methods, where we describe our method as *forward SDE-free*.

We summarize our optimization-free and forward SDE-free approach for learning the score function in the following Algorithm 10. Then later in Section 5.3 and Section 5.4, we elaborate on the implementational details of Algorithm 10.

Algorithm 10 Optimization-free Diffusion Model

Input: Initial data $\{x^{(j)}\}_{j=1}^{N}$ sampled from ρ_0. Time-grid $t_{\text{grid}} \in [0, T]$.

Select base distribution: Select a base distribution in (5.3).

Compute eigenfunction: Compute associated eigenfunctions of (5.11) and choose a set of eigenfunctions $\{f_k\}_{k\in\mathcal{S}}$ for score representation.

Solve for Score: Compute integral $\{\int f_k(x)\rho_0(x)\mathrm{d}x\}_{k\in\mathcal{S}'}$ via Monte Carlo integration $\{\frac{1}{N}\sum_{j=1}^{N} f_k(x^{(j)})\}_{k\in\mathcal{S}'}$. $\mathcal{S}'$ is the set for expansion in (5.14).

for $t \in t_{\text{grid}}$ **do**

    Compute coefficient $A(t)$ in (5.15) and $B(t)$ in (5.16).

    Solve linear system in (5.10).

    Compute score function $s(t, x)$ in (5.5).

end for

Output: Score function $\{s(t, x)\}_{t\in t_{\text{grid}}}$.

One of the key tasks in Algorithm 10 is to choose the base distribution and a rich but small subset of the corresponding eigenbasis. Furthermore, efficiently solving the score is important to make the algorithm feasible for high-dimensional problem settings. In the following, we will give some examples of base distributions and the corresponding eigenbasis. Crucially, in order to make the spectral decomposition of the score computationally accessible in high dimensions, one needs to go beyond smoothness assumptions and takes other hidden structures into account. To this end, we will exploit so-called locality assumptions by expanding the score in a cluster basis.

5.3 Application of Main Algorithm: Single-variable Distribution

We will start by exemplifying the necessary steps of Algorithm 10 in the case of single-variable distributions focusing on potential base distributions, eigenfunctions and the complexity of

the optimization problem. All notions introduced here are also used in the multivariate case and the single-variable setting is only to have a clean presentation of the ideas.

Examples for Base Distributions

The most commonly used base distribution in diffusion models is a *Gaussian distribution*, described by the potential $V(x) = \frac{1}{2}\alpha x^2$. The associated overdamped Langevin dynamic is the Ornstein-Uhlenbeck process and the associated eigenfunctions are (properly scaled) Hermite polynomials.

Alternatively, we consider a *uniform distribution* over the compact domain $[-L, L]$. It turns out that by choosing the potential $V(x) = 1$ and imposing periodic boundary conditions for the diffusion process, i.e. a periodic Brownian motion, the stationary measure of the associated Langevin dynamics is the uniform distribution. The corresponding eigenfunctions are trigonometric polynomials.

Further computation details can be found in [174].

5.3.1 Choosing a good Ansatz based on Eigenfunctions

After choosing the base distribution and obtaining the associated eigenfunctions, we discuss how to select an appropriate subset of eigenfunctions to achieve an accurate score representation. Recall that the expansion in (5.13) decays exponentially in time at a rate determined by $|\lambda_k|$, where all eigenvalues of the operator $\mathcal{L}$ in (5.11) are non-positive. Based on this observation, we sort the eigenfunctions in order of increasing $|\lambda_k|$ and prioritize those with smaller magnitudes, as they contribute more substantially to the representation. Thus, we choose the index set $\mathcal{S}$ for eigenfunctions to include the $|\mathcal{S}|$ eigenfunctions corresponding to the eigenvalues with smallest magnitudes.

5.3.2 Computational Complexity

The key tasks in algorithm 10 is to solve the linear system to recover an approximation to the score function. Building the coefficients of A and b for time $t = 0$ is straightforward through Monte Carlo integration which requires $O(N|\mathcal{S}'|)$ operations. To obtain the integrals at later times we do the expansion in (5.14) and then compute $A(t)$ and vector $b(t)$ via (5.15) and (5.16). For the two eigenfunction choices discussed above—Hermite polynomials and Fourier basis—the expansions in (5.14) can hold exactly by using $2|\mathcal{S}|$ eigenfunctions with the smallest magnitudes for the expansion ($|\mathcal{S}'| = 2|\mathcal{S}|$): both the product of two eigenfunctions and the derivative of an eigenfunction can be expressed as linear combinations of the basis functions. Moreover, since $V'(x) = \alpha x$ in the Hermite polynomial setting and $V'(x) = 0$ in the Fourier basis setting, there is no interpolation error introduced in representing the potential gradient in either case. Thus, computing the coefficients for the expansions in (5.14) involves $O(|\mathcal{S}|^3)$ operations.

For each time step t, evaluating $A(t)$ and $b(t)$ as in (5.15) and (5.16) also requires $O(|\mathcal{S}|^3)$ operations. Solving the resulting linear system (5.10) (Line 7 of Algorithm 10) using standard methods likewise costs $O(|\mathcal{S}|^3)$. Assuming a total of N_t time steps, the overall cost for these computations scales as $O(N_t|\mathcal{S}|^3)$. In summary, the total computational complexity of our method is $O(N|\mathcal{S}| + N_t|\mathcal{S}|^3)$.

Notably, our approach requires only a single Monte Carlo simulation in Line 4 of Algorithm, which contrasts sharply with the standard Euler-Maruyama scheme. In the latter, the integrals in $A(t)$ and $b(t)$ are computed by samples x_t generating at each time step t. This results in a total computational cost of $O(NN_t|\mathcal{S}|^2 + N_t|\mathcal{S}|^3)$, which can be significantly higher than that of our method, particularly in large-sample settings.

5.4 Application of Main Algorithm: Multi-variable Distribution

The choice of the basis set and the numerical implementation as exemplified in the previous section does not readily extend to the higher dimensional setting without accounting for hidden structures that mitigate the curse of dimensionality. Especially choosing the set $\mathcal{S}$, i.e. fixing our basis, is essential and needs to be done carefully in order for it to be small enough to be computationally feasible but rich enough to retain enough expressive power to actually approximate the score function. To that end, we will assume that the density ρ_0 is a perturbation of the base distribution ρ_∞.

5.4.1 Mean-field Approximations as Base Distribution

In this subsection, we focus on the choice of base distribution in the high-dimensional setting. In general, we want to choose a distribution that is easy to sample from, e.g. seperable densities. One can simply pick a Gaussian distribution with diagonal covariance matrix or a uniform distribution on a hypercube. We, however, propose a novel base distribution based on a mean-field approximation of the underlying data distribution, constructed directly from empirical samples.

While the true underlying density ρ_0 is typically not mean-field, we can construct a mean-field approximation of the form $\rho_{\mathrm{MF}}(x) = \prod_{i=1}^{d} \rho_i(x_i)$. This approximation enables efficient i.i.d. sampling. To determine each one-dimensional marginal $\rho_i(x_i)$, we match its first few moments to those of underlying density ρ_0. Specifically, for a fixed moment order n_{m}, we impose the constraint $\mu_{i,j} = \int x_i^j \rho_0(x)\,\mathrm{d}x$ for $0 \leq j \leq n_{\mathrm{m}}$ where the right-hand side is estimated via Monte Carlo integration using samples drawn from ρ_0. Each marginal ρ_i is

then obtained by solving the following entropy maximization problem:

$$\begin{aligned} &\min_{\rho_i} \int \rho_i(x_i) \log \rho_i(x_i) \, \mathrm{d}x_i \\ \text{subject to} \quad &\rho_i(x_i) \geq 0, \quad \int x_i^j \rho_i(x_i) \, \mathrm{d}x_i = \mu_{i,j}, \quad \text{for } 0 \leq j \leq n_{\mathrm{m}}. \end{aligned} \tag{5.17}$$

In order to solve the constrained optimization problem, we define the associated Lagrangian

$$L(\rho_i, \{\nu_j^{(i)}\}_j) := \int \rho_i(x_i) \log(\rho_i(x_i)) \mathrm{d}x_i + \sum_{j=0}^{n_{\mathrm{m}}} \nu_j^{(i)} (\int x_i^j \rho_i(x_i) \mathrm{d}x_i - \mu_{i,j}).$$

The dual variables $\nu_j^{(i)}$ can be computed via standard numerical optimization approach [19, 224] and ρ_i are obtained through these dual variables via the first order necessary conditions

$$\frac{\delta L(\rho_i, \{\nu_j^{(i)}\}_j)}{\delta \rho_i} = 0 \implies \rho_i(x_i) = \exp(-\sum_{j=0}^{n_{\mathrm{m}}} \nu_j^{(i)} x_i^j - 1).$$

After getting $\{\rho_i\}_{i=1}^d$ that defines ρ_{MF}, we let $\rho_\infty = \rho_{\mathrm{MF}}$, which serves as the high-dimensional base distribution.

5.4.2 Clusters of Eigenfunctions

For our proposed method, the expansion of the score function in terms of eigenfunctions associated to the base distribution is crucial. For non-standard, seperable Gibbs measures, we need to approximate the eigenfunctions numerically as follows. Let $\rho_\infty(x) \propto \exp(-V(x)) = \exp(-\sum_{i=1}^d V_i(x_i))$, where each V_i is a single variable function. In this case, the high-dimensional backward Kolmogorov operator can be decomposed as the sum of one-dimensional operators

$$\mathcal{L} = -\nabla V(x) \cdot \nabla + \beta^{-1}\Delta = \sum_{i=1}^{d}(-\frac{\partial V(x)}{\partial x_i}\frac{\partial}{\partial x_i} + \beta^{-1}\frac{\partial^2}{\partial x_i^2})$$
$$= \sum_{i=1}^{d}(-\frac{\partial V_i(x)}{\partial x_i}\frac{\partial}{\partial x_i} + \beta^{-1}\frac{\partial^2}{\partial x_i^2}) := \sum_{i=1}^{d}\mathcal{L}_i. \tag{5.18}$$

Let $\lambda_{n_i}^{(i)}$ and $\phi_{n_i}^{(i)}(x_i)$ denote the eigenvalues and eigenfunctions of the one-dimensional operator $\mathcal{L}_i$. Then, the eigenvalues and eigenfunctions of the full high-dimensional operator $\mathcal{L}$ are given by $\sum_{i=1}^{d}\lambda_{n_i}^{(i)}$ and $\prod_{i=1}^{d}\phi_{n_i}^{(i)}(x_i)$, respectively. These eigenfunctions can be constructed by solving individual 1D eigenvalue equation with a finite difference discretization.

For the next step, we demonstrate how to select a subset of eigenfunctions for score representation. In the single-variable case, we select the eigenfunctions with small eigenvalues, which typically gives rise to smooth approximations. However, one cannot directly extend this strategy of choosing eigenvalues lesser than a certain threshold, directly to the high-dimensional setting. The reason is that such a choice corresponds to selecting the interior of a ball, with a weighted norm induced by the eigenvalues, in $\mathbb{Z}^d$ with a volume that scales exponentially in d. Therefore, it is crucial to make a well-informed choice of subsets of these balls by adopting different strategies to select the eigenfunctions. We, thus, consider a structured subset of eigenfunctions known as the *j-cluster* basis [37, 175], which consists of all j-variable basis functions:

$$\mathcal{B}_{j,d} = \left\{ \phi_{n_{i_1}}^{(i_1)}(x_{i_1}) \cdots \phi_{n_{i_j}}^{(i_j)}(x_{i_j}) \;\middle|\; 1 \le i_1 < \cdots < i_j \le d,\ 1 \le n_{i_1}, \ldots, n_{i_j} \le n \right\}, \tag{5.19}$$

where n represents the number of one-dimensional eigenfunctions for each dimension. Each eigenfunction in $\mathcal{B}_{j,d}$ involves at most a product of j one-dimensional eigenfunctions. This approach can reduce the total number of basis functions from n^d to $\binom{d}{j}n^j$, offering substantial

computational savings while preserving expressive power.

To further reduce the computational complexity, we introduce two assumptions on the underlying density ρ_0:

- We assume that a 2-cluster basis is sufficient to represent the score function. This assumption seems plausible, if the target density ρ_0 takes a high-dimensional Gibbs measure of the form $\exp(-V(x))$, where $V(x)$ is at most a 2-cluster function.

- Additionally, we assume that it suffices to use a sparse set of 2-cluster basis to represent the score function. This is possible, if the potential $V(x) = \sum_{(i,j)} V_{ij}(x_i, x_j)$ for the target Gibbs measure only couples a sparse set of (i, j) pair. This type of potentials are quite common in physics application where there is near-sightedness.

In Example 3 in Appendix D.2 we demonstrate how such structures transfer from the potential to the score function in the perturbative regime. These structural assumptions allow us to expand the score function using a local 2-cluster basis, where basis functions are restricted to interactions between nearby coordinates:

$$\mathcal{B}_{2,d}^{\text{local}} = \left\{ \phi_{n_i}^{(i)}(x_i)\, \phi_{n_{i'}}^{(i')}(x_{i'}) \;\middle|\; 1 \le i < i' \le d,\ i' \le i + d_{\text{b}},\ 1 \le n_i, n_{i'} \le n \right\}, \tag{5.20}$$

where d_{b} is a dimension-independent bandwidth parameter that controls the locality of interactions. Under this formulation, when the score function lies within $\mathcal{B}_{2,d}^{\text{local}}$, the number of coefficients in $c^{(i)}(t)$ is reduced to at most $|\mathcal{S}| \le d_{\text{b}} d n^2$, resulting in a representation whose complexity grows linearly with the dimension d.

5.4.3 Fast Tricks in High Dimensions

Let us now review the computational complexity of solving the linear system exploiting the local 2-cluster basis $\mathcal{B}_{2,d}^{\text{local}}$ in (5.20). We summarize the key components of the method and

focus on fast numerical techniques, detailed computational steps for (5.15) and (5.16) are provided in Appendix D.1.

In the initial step (Line 4 of Algorithm 10), we compute integrals of each pair of basis functions in $\mathcal{B}_{2,d}^{\text{local}}$ against $\rho_0(x)$ via Monte Carlo integration using its samples $\{x^{(j)}\}_{j=1}^N$. This step incurs a computational cost of $O(Nd_{\text{b}}^2 d^2 n^4)$. Next, we compute the coefficients for the expansions in (5.14), which serve as a preparatory step for efficiently evaluating $A(t)$ and $b(t)$ in (5.8). We use the same choice of $\mathcal{S}'$ as in Section 5.3.2 for the expansions. This step requires $O(d_{\text{b}} d n^3)$ operations. For each time step t, the computation of $A(t)$ and $B(t)$ in (5.8) costs $O(d_{\text{b}}^2 d^2 n^4 + d_{\text{b}} d^2 n^3) = O(d_{\text{b}}^2 d^2 n^4)$. Solving the linear system (5.10) using a standard solver requires $O(d_{\text{b}}^3 d^3 n^6)$ operations. Therefore, the overall computational complexity of the method is $O(Nd_{\text{b}}^2 d^2 n^4 + N_t d_{\text{b}}^3 d^3 n^6)$ where N_t denotes the total number of time steps.

Moreover, in the high-dimensional settings, the matrix $A(t)$ in (5.8) is often ill-conditioned, and directly inverting it can result in an unstable score estimator. To address this issue, a common and effective strategy is to apply singular value thresholding: we specify a threshold and discard all singular values of $A(t)$ that fall below it, then compute the pseudoinverse using the remaining components. An alternative approach is to incorporate a regularization term, such as ℓ_2 regularization, when solving the linear system (5.10). In our experiments, we observe that $A(t)$ frequently exhibits a low-rank structure, and thus we adopt the thresholding approach in our method.

One dominant term in the computational complexity described above $O(N_t d_{\text{b}}^3 d^3 n^6)$ arises from solving the linear system (5.10). Fortunately, by exploiting the low-rank structure of $A(t)$ discussed above, this cost can be significantly reduced using randomized low-rank approximation techniques [133, 82]. The key idea is that a low-rank approximation can be constructed by accurately capturing the dominant left singular subspace of the matrix, without explicitly reconstructing its full column space. Specifically, we can generate a random Gaussian matrix $\Omega \in \mathbb{R}^{|\mathcal{S}| \times \tilde{r}_A}$ with $\tilde{r}_A \ll |\mathcal{S}|$, typically independent of the problem dimension,

and we then estimate its left singular subspace using the reduced matrix $A(t)\Omega$:

$$[U(t), \Sigma(t), V(t)] = \mathrm{SVD}(A(t)\Omega, r_A),$$

where $\mathrm{SVD}(A(t)\Omega, r_A)$ denotes a truncated SVD with rank r_A, and $U(t) \in \mathbb{R}^{|\mathcal{S}| \times r_A}$ is the resulting orthonormal basis. The rank r_A is typically smaller than $\tilde{r}_A$ and can be adaptive to the problem. The matrix $U(t)U(t)^T A(t)$ then serves as a low-rank approximation to $A(t)$, computed at a cost of $O(|\mathcal{S}|^2 \tilde{r}_A)$. The corresponding linear system can be approximated by projecting onto this low-rank subspace:

$$U(t)U(t)^T A(t)C(t) = -B(t) \quad \Rightarrow \quad \Big(U(t)^T A(t)\Big)\, C(t) = -U(t)^T B(t), \tag{5.21}$$

where $U(t)^T A(t)$ is a matrix of size $r_A \times |\mathcal{S}|$, significantly smaller than the original system. Solving this reduced system requires only $O(|\mathcal{S}|^2 \tilde{r}_A + |\mathcal{S}|^2 r_A + |\mathcal{S}| d r_A + |\mathcal{S}| r_A^2) = O(|\mathcal{S}|^2 \tilde{r}_A)$ operations, in contrast to the classical $O(|\mathcal{S}|^3)$ cost for direct inversion.

By incorporating this fast numerical technique, the overall computational complexity is reduced to

$$O(N d_{\mathrm{b}}^2 d^2 n^4 + N_t d_{\mathrm{b}}^2 d^2 n^6), \tag{5.22}$$

with assuming a sketch size $\tilde{r}_A = O(n^2)$, which depends only on n. In contrast, for conventional Euler-Maruyama simulation, the dominant computational cost is $O(N N_t d_{\mathrm{b}}^2 d^2 n^4)$, which is significantly higher in large-sample regimes.

5.5 Error Estimation in the Perturbative Regime

In this section, we show that if the underlying density is a perturbation of a mean-field Gibbs measure, the approximation error can be bounded in terms of the perturbation scale. We begin by formally stating and explaining the assumptions used in the analysis.

Let ρ_0 denote the underlying density from which we wish to sample and assume it takes the following form:

$$\rho_0(x) = C_\rho \, \rho_\infty(x) \left(1 + \delta \sum_{i=1}^{\infty} p_i f_i(x)\right), \tag{5.23}$$

where $f_i(x)$ are the eigenfunctions associated with the base distribution ρ_∞, p_i are the corresponding coefficients, and δ controls the perturbation scale. The constant C_ρ ensures proper normalization. The second term in the formula represents the deviation of ρ_0 from ρ_∞. In the following, we state the assumptions placed on the parameters p_i and δ to facilitate our analysis.

Assumption 7. *We assume the base distribution takes the form $\rho_\infty(x) = \frac{1}{Z} e^{-V(x)}$ for some analytic, coercive potential V such that the eigenvalues $\{\lambda_i\}$ of the associated operator $\mathcal{L}$ satisfy $0 = \lambda_0 > \lambda_1 \geq \lambda_2 \geq \dots$. Let p_i satisfy*

$$\sup_{x \in \mathbb{R}^d} |\sum_{i=1}^{\infty} p_i f_i(x)| \leq 1, \tag{5.24}$$

and assume that there is a non-decreasing function $r : \mathbb{Z}_+ \to \mathbb{R}_+$ with $\lim_{M \to \infty} r(M) = 0$ such that

$$\int_{\mathbb{R}^d} \left\| \nabla \left(\sum_{i=M}^{\infty} p_i f_i(x) \right) \right\|^2 \rho_0(x) \mathrm{d}x \leq r(M). \tag{5.25}$$

To justify the assumption, we provide examples related to the Ornstein-Uhlenbeck process and the periodic Brownian motion respectively, detailed in Appendix D.2.

These assumptions allow us to analyze the score estimation error in the perturbative regime. Specifically, we will demonstrate how the choice of a finite-dimensional sub-eigensystem will affect the accuracy of the estimated score function. To this end, we first decompose the total error into two components: an approximation error and a statistical error.

Theorem 8. *Let $F_M = \{f_1, \ldots, f_M\}$ be our hypothesis class and let s_M be minimizing*

$$\int_0^\infty \int_{\mathbb{R}^d} \left(\frac{1}{2} \|s_M(t,x)\|^2 + \operatorname{div}(s_M(t,x)) \right) \rho_t(x) \mathrm{d}x \mathrm{d}t \tag{5.26}$$

over F_M. Now let $\{x^{(j)}\}_{j=1}^N$ be N i.i.d. samples drawn from ρ_0. Assume that for every $\mathrm{err}_{\mathrm{stat}} > 0$ with high probability $\mathbb{P} = \mathbb{P}(N, \mathrm{err}_{\mathrm{stat}}, F_M)$, depending on N, $\mathrm{err}_{\mathrm{stat}}$ and the (Rademacher) complexity of the model class F_M, it holds

$$\left| \int_0^\infty \int_{\mathbb{R}^d} \left(\frac{1}{2} \|s(t,x)\|^2 + \operatorname{div}(s(t,x)) \right) \rho_t(x) \mathrm{d}x \mathrm{d}t \right.$$
$$\left. - \frac{1}{N} \sum_{j=1}^N \int_0^\infty \frac{1}{2} \|s(t, x^{(j)}(t))\|^2 + \operatorname{div}(s(t, x^{(j)}(t))) \mathrm{d}t \right| \leq \mathrm{err}_{\mathrm{stat}}$$

for all $s \in F_M$. Let $\{x^{(j)}(t)\}_{j=1}^N$ be the trajectories of $\{x^{(j)}\}_{j=1}^N$ under the overdamped Langevin dynamics induced by the base distribution. Suppose $s(t,x) = \nabla \log \rho_t(x)$ is the ground truth score function and $s_{M,N}$ be the minimizer of

$$\frac{1}{N} \sum_{j=1}^N \int_0^\infty \frac{1}{2} \|s_{M,N}(t, x^{(j)}(t))\|^2 + \operatorname{div}(s_{M,N}(t, x^{(j)}(t))) \mathrm{d}t.$$

Then we have with probability $\mathbb{P}(N, \mathrm{err}_{\mathrm{stat}}, F_M)$ that

$$\left| \int_0^\infty \int_{\mathbb{R}^d} \left(\frac{1}{2} \|s_{M,N}(t,x)\|^2 - \frac{1}{2} \|s(t,x)\|^2 + \operatorname{div}[s_{M,N}(t,x) - s(t,x)] \right) \rho_t(x) \mathrm{d}x \ \mathrm{d}t \right|$$
$$\leq 2\mathrm{err}_{\mathrm{stat}} + \mathrm{err}_{\mathrm{appr}},$$

where

$$\begin{aligned}\mathrm{err}_{\mathrm{appr}} &= |\int_0^\infty \int_{\mathbb{R}^d} \left(\frac{1}{2}\|s_M(t,x)\|^2 - \frac{1}{2}\|s(t,x)\|^2 + \mathrm{div}(s_M(t,x) - s(t,x)\right)\rho_t(x)\mathrm{d}x\mathrm{d}t| \\ &= \int_0^\infty \int_{\mathbb{R}^d} \frac{1}{2}\|s_M(t,x) - \nabla \log \rho_t(x)\|^2 \rho_t(x)\mathrm{d}x\mathrm{d}t.\end{aligned}$$

The proof follows directly from the triangle inequality and integration by parts. The following theorem provides an approximation error analysis based on the perturbation assumption 7.

Theorem 9. *Let $M \in \mathbb{N}$. Let $f_1, \ldots, f_M$ be the eigenfunctions of ρ_∞ and Assumption 7 be fulfilled. Assume that the potential $V \in F_M$. Then the following bound holds*

$$\min_{s_M \in L^2([0,\infty), F_M)} \int_0^\infty \int_{\mathbb{R}^d} \frac{1}{2}\|\nabla \log \rho_t(x) - s_M(t,x)\|^2 \rho_t(x)\mathrm{d}x\, \mathrm{d}t \leq \frac{\delta^2}{|\lambda_1|}(r(M+1) + (\frac{\delta}{1-\delta})^2 r(1)). \tag{5.27}$$

Proof. Using eigenfunctions $\{f_i\}$ and Assumption 7, we have the following representation for time-dependent density $\rho_t(x)$:

$$\begin{aligned}\rho_t(x) &= e^{\mathcal{L}^* t}\rho_0(x) = C_\rho e^{\mathcal{L}^* t}\left(\rho_\infty\left(1 + \delta \sum_{i=1}^\infty p_i f_i(x)\right)\right) \\ &= C_\rho \rho_\infty \left(1 + \delta \sum_{i=1}^\infty p_i e^{\lambda_i t} f_i(x)\right) = \frac{C_\rho}{Z} e^{-V(x)}\left(1 + \delta \sum_{i=1}^\infty p_i e^{\lambda_i t} f_i(x)\right)\end{aligned}$$

Therefore,

$$\begin{aligned}\nabla \log(\rho_t(x)) &= -\nabla V(x) + \nabla \log(1 + \delta \sum_{i=1}^\infty p_i e^{\lambda_i t} f_i(x)) \\ &= -\nabla V(x) + \frac{\delta \sum_{i=1}^\infty p_i e^{\lambda_i t} \nabla f_i(x)}{1 + \delta \sum_{i=1}^\infty p_i e^{\lambda_i t} f_i(x)}.\end{aligned}$$

Recall that $\frac{1}{1+y} = \sum_{k=0}^{\infty}(-1)^k y^k$ for $|y| < 1$. With Assumption 7, we get

$$|\sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)| \leq |\sum_{i=1}^{\infty} p_i f_i(x)| \leq 1$$

Therefore, we can expand the second term:

$$\begin{aligned}\nabla \log \rho_t(x) = -\nabla V(x) + \delta \sum_{i=1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x) \\ + \left(\delta \sum_{i=1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x)\right) \sum_{k=1}^{\infty} (-1)^k \left(\delta \sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)\right)^k\end{aligned}$$

Thus by choosing

$$s_M(t,x) = -\nabla V(x) + \delta \sum_{i=1}^{M} p_i e^{\lambda_i t} \nabla f_i(x) \in L^2([0,\infty), F_M),$$

we have

$$\begin{aligned}&\|\nabla \log \rho_t(x) - s_M(t,x)\|^2 \\ =&\left\|\delta \sum_{i=M+1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x) + \left(\delta \sum_{i=1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x)\right) \sum_{k=1}^{\infty} (-1)^k \left(\delta \sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)\right)^k \right\|^2 \\ \leq& 2\delta^2 \left[\left\|\sum_{i=M+1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x)\right\|^2 + \left\|\sum_{i=1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x) \sum_{k=1}^{\infty} (-1)^k \left(\delta \sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)\right)^k \right\|^2\right] \\ \leq& 2\delta^2 \left[\left\|\sum_{i=M+1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x)\right\|^2 + \left\|\sum_{i=1}^{\infty} p_i e^{\lambda_i t} \nabla f_i(x) \sum_{k=1}^{\infty} \delta^k \right\|^2\right] \\ \leq& 2\delta^2 e^{2\lambda_1 t} \left[\left\|\sum_{i=M+1}^{\infty} p_i \nabla f_i(x)\right\|^2 + \frac{\delta^2}{(1-\delta)^2} \left\|\sum_{i=1}^{\infty} p_i \nabla f_i(x)\right\|^2\right].\end{aligned}$$

The first inequality is from Cauchy inequality. For the last inequality, all eigenvalues λ_i are

non-positive, so $e^{\lambda_1 t}$ is the largest coefficient among $\{e^{\lambda_i t}\}_{i=1}^{\infty}$. Therefore,

$$\begin{aligned}
&\int_0^\infty \int \|\nabla \log \rho_t(x) - s_M(t,x)\|^2 \rho_t(x) \mathrm{d}x\mathrm{d}t \\
&\leq \frac{\delta^2}{|\lambda_1|} \left(\int \|\nabla \sum_{i=M+1}^{\infty} p_i f_i(x)\|^2 \rho_0(x) \mathrm{d}x + (\frac{\delta}{1-\delta})^2 \int \|\nabla \sum_{i=1}^{\infty} p_i f_i(x)\|^2 \rho_0(x) \mathrm{d}x \right) \\
&\leq \frac{\delta^2}{|\lambda_1|} (r(M+1) + (\frac{\delta}{1-\delta})^2 r(1))
\end{aligned}$$

where we use the fact $\int_0^\infty e^{2\lambda_1 t} \mathrm{d}t = \frac{1}{-2\lambda_1} = \frac{1}{2|\lambda_1|}$. □

This result shows that the approximation error is closely tied to the perturbation scale δ: the error vanishes as $\delta \to 0$. However, even as the number of basis functions M tends to infinity, a small residual term in the approximation error may still persist. A more refined analysis—leveraging certain algebraic structures of the basis—can strengthen this result. We will illustrate this in the context of one-dimensional periodic Brownian motion.

Theorem 10. *Let $f_i(x) = e^{\sqrt{-1}\omega i x}$ and λ_i as the corresponding eigenvalue, $i \in \mathbb{Z}$. Assume coefficient $|p_i| \leq \gamma^i$ for some $\gamma < 1$. Then for any $\varepsilon > 0$, there is M and a $s_{2M} \in \mathrm{Span}\{f_{-2M}, \ldots, f_{2M}\}$ such that*

$$\int_0^\infty \int (\partial_x \log \rho_t(x) - s_{2M}(t,x))^2 \rho_t(x) \mathrm{d}x\mathrm{d}t \leq \frac{\varepsilon}{|\lambda_1|} \tag{5.28}$$

This theorem shows that the approximation error can be made arbitrarily small, provided that the number of basis functions M is sufficiently large. The corresponding proof is provided in the appendix D.3.

Furthermore, we provide a result towards the statistical error via defining a space for score function:

$$\mathcal{H}_S = \Big\{ s(t,x) \;\Big|\; s^{(i)}(t,x) = \big\langle c^{(i)}(t), f(x) \big\rangle - \partial_{x_i} V(x), \quad c^{(i)}(t) \in \mathbb{R}^{|\mathcal{S}|}, \\ \|C(t)\|_F \le C_M, \quad \forall\, t \in [0,T],\ 1 \le i \le d \Big\}. \tag{5.29}$$

where $C(t) = [c^{(1)}(t), c^{(2)}(t), \cdots, c^{(d)}(t)]$. $|\mathcal{S}|$ is the number of basis for this span and C_M is the upper bound of norm of coefficient $\|C(t)\|_F$.

Theorem 11. *There exists a constant $C_M > 0$ such that $s_{M,N}(t,x) \in \mathcal{H}_S$ for all (t,x), and its statistical error satisfies*

$$\mathrm{err}_{\mathrm{stat}} \;\le\; \left(\frac{1}{2} C_M^2 M \;+\; C_M M \;+\; \sqrt{dM} \right) \frac{C_0}{|\lambda_1| \sqrt{N}}. \tag{5.30}$$

Here C_0 is a constant depending only on the chosen basis, N is the sample size.

This theorem shows that the statistical error decays at the Monte Carlo rate in the sample size, with a reasonable dependence on the complexity of the hypothesis space. The proof is provided in Appendix D.4.

5.6 Numerical Results

In this section, we evaluate our approach on high-dimensional Boltzmann distributions and the MNIST data set. We begin by summarizing the overall numerical procedure. In the first step, we estimate the score function using Algorithm 10 over an uniform time grid $t_{\mathrm{grid}} \subset [0,T]$, where T is chosen to be sufficiently large so that the initial data are fully diffused to the base distribution (5.3). In the second step, we generate samples from the base distribution using the Metropolis–Hastings algorithm. Since our choices of base distributions are mean-field, this sampling step is relatively simple. In the third step, we simulate the reverse-time SDE (5.4) via Euler-Maruyama scheme over t_{grid}, starting from the generated samples at time $t = T$ and evolving them backward to $t = 0$. The resulting samples form an empirical distribution that approximates the target density ρ_0. We use the discrepancy

between this empirical distribution and the ground truth distribution as the primary measure of error in our evaluation.

5.6.1 One-dimensional Double-well Density

We start our numerical experiments with a one-dimensional double-well density. A double-well density refers to a probability distribution whose potential energy function exhibits two distinct minima, or "wells," separated by a barrier. This structure often arises in physical systems, such as in statistical mechanics or molecular dynamics. The corresponding density is defined as

$$\rho_0(x) = \frac{1}{Z}\mathrm{e}^{-\beta_{\mathrm{DW}}(1-x^2)^2/4},$$

where β_{DW} is the inverse of temperature and Z is the normalization constant. We set $\beta_{\mathrm{DW}} = 2.0$ for our test. In this experiment, we consider two choices of eigenfunctions for representing the score function introduced in Section 5.3: Hermite polynomials and the Fourier basis. We generate 40,000 initial samples from ρ_0 and evaluate the accuracy of score estimation using the relative L_2 score error:

$$\frac{\|s(t,x) - s^*(t,x)\|_{L_2(\rho_t)}}{\|s^*(t,x)\|_{L_2(\rho_t)}},$$

where $s^*(t,x)$ denotes the ground truth score function, obtained by solving the one-dimensional Fokker-Planck equation (5.12) with finite difference method. This error is estimated using Monte Carlo integration with 100,000 samples from ρ_t via the Euler-Maruyama scheme.

Figure 5.1 presents the results using Hermite polynomials as the basis. We fix potential $V(x) = \frac{1}{2}x^2$ and set the total diffusion time to $T = 2.0$ and time step $\Delta t = 0.002$. Note that the initial score function is given by $s(0,x) = \partial_x \log \rho_0(x) = -\beta_{\mathrm{DW}} x(x^2 - 1)$, which is a cubic polynomial. This structure makes Hermite polynomials a natural and effective choice for the score representation.

The left subfigure of Figure 5.1 shows the relative score error as a function of the number of basis functions, with $\beta = 1.0$ fixed. The error remains stable as the basis size increases, indicating good approximation properties. The middle subfigure examines the effect of varying β, which controls the standard deviation of the Gaussian base distribution. The results remain consistent across different β values, demonstrating the robustness of our approach. Finally, the right subfigure illustrates the evolution of the score function over time for number of eigenfunctions $n = 9$ and $\beta = 1.0$, showing a transition from a cubic polynomial toward a linear function, consistent with the behavior expected under Ornstein–Uhlenbeck dynamics.

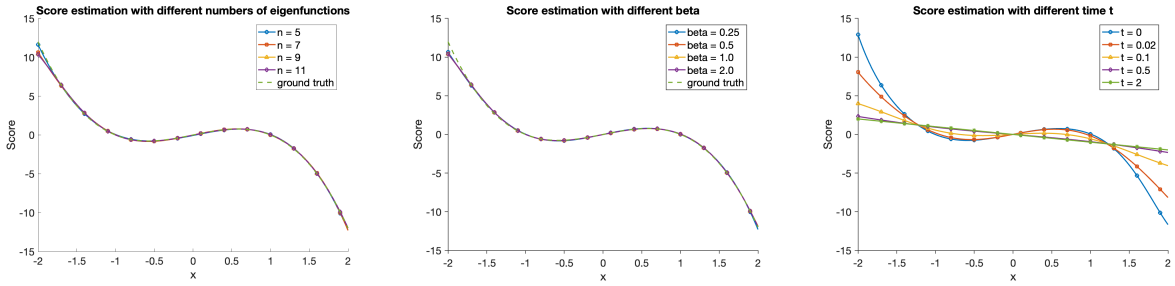


Figure 5.1: One-dimensional score estimation using Hermite polynomials. (Left) Estimated score functions for different numbers of eigenfunctions $n = 5, 7, 9, 11$. The corresponding relative errors are 0.0412, 0.0408, 0.0405, and 0.0401, respectively. (Middle) Score estimates for varying values of the parameter β, which controls the standard deviation of the Gaussian base distribution. The relative errors for $\beta = 0.25, 0.5, 1.0, 2.0$ are 0.0428, 0.0432, 0.0421, and 0.0421, respectively. (Right) Temporal evolution of the score function for $t = 0, 0.02, 0.1, 0.5, 2$. As expected, the score converges to a linear function over time in the Hermite polynomial representation.

Next, we present results using the Fourier basis, with parameters fixed as $\beta = 0.5$, $T = 2.0$ and time step $\Delta t = 0.002$. The left subfigure of Figure 5.2 shows how the relative score error varies with the number of basis functions. As expected, the error is significantly higher when the number of basis functions is insufficient to capture the complexity of the score function. The middle subfigure illustrates the relative score error as a function of the domain size L. While the Fourier basis is periodic over the interval $[-L, L]$, the ground truth score function, defined on the support of the initial data distribution, is not inherently periodic. To ensure that the fitted score function accurately approximates the ground truth within

the relevant region, the domain $[-L, L]$ must be large enough to cover the support of the initial distribution. This explains the poor performance when $L = 2$, as the domain is too small to capture the full support. The right subfigure visualizes the evolution of the score function over time. As time increases, the score function gradually converges to a constant, consistent with the expected long-term behavior under the chosen dynamics.

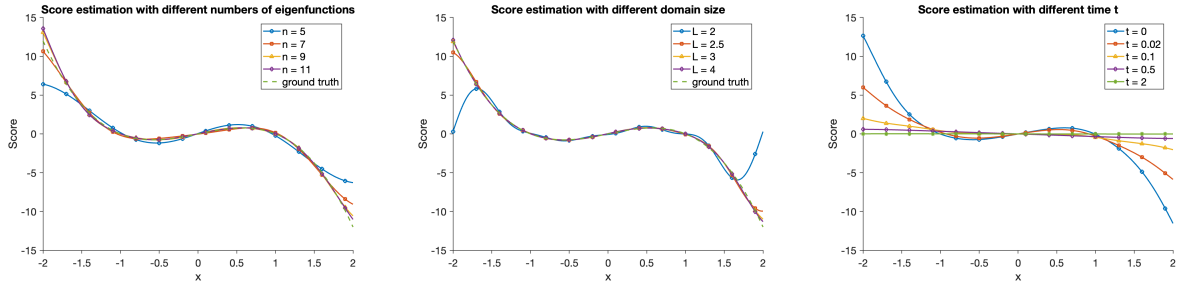


Figure 5.2: One-dimensional score estimation using the Fourier basis. (Left) We vary the number of eigenfunctions and visualize the estimated score. The relative score errors are 0.2756, 0.1053, 0.0666, and 0.0502 for $n = 5, 7, 9, 11$, respectively. (Middle) We vary the domain size in the Fourier basis and plot the resulting score functions. The corresponding relative score errors are 0.3612, 0.0763, 0.0501, and 0.0414 for $L = 2, 2.5, 3, 4$, respectively. (Right) We show the evolution of the score function over time for $t = 0, 0.02, 0.1, 0.5, 2$. As expected, the score function in the Fourier representation converges toward a constant zero function as time increases.

5.6.2 High-dimensional Double-well Density

In this subsection, we consider the high-dimensional double-well density

$$\rho_0(x) = \frac{1}{Z} \mathrm{e}^{-\beta_{\mathrm{DW}} V_{\mathrm{DW}}(x)}, V_{\mathrm{DW}}(x) = \sum_{i=1}^{d} \frac{1}{4}(1 - x_i^2)^2,$$

where β_{DW} is the inverse of temperature and Z is the normalization constant. The density is one type of mean-field density since the variables do not interact with each other, providing a simple high-dimensional example to illustrate the performance of our approach. We test $d = 32$ and $\beta_{\mathrm{DW}} = 8$. In this setting, we use $n = 10$ eigenfunctions per dimension and set the bandwidth to $d_{\mathrm{b}} = 2$ for the local cluster basis representation (5.20). The score function

is estimated over the interval $t \in [0, T]$ with $T = 2$ and time-step $\Delta t = 0.002$, using 40,000 initial samples.

We evaluate our method using all three types of basis functions. For the Hermite polynomial basis, we fix $\beta = 1.0$ and $V(x) = \frac{1}{2}\|x\|_2^2$ in (5.2). For the Fourier basis, we choose $L = 5.0$ and $\beta = 0.25$ in (5.2), since the dynamics converge slowly in this setting, and a smaller β can accelerate convergence. For the mean-field approximated basis, we fix $\beta = 1.0$ and construct the approximating density by matching the first 6 moments in the entropy maximization problem (5.17).

After learning the score function, we generate 40,000 samples from the base distribution and simulate the reverse-time SDE with time-step $\Delta t = 0.002$. For visualization, we primarily present one-dimensional marginal plots comparing the generated samples with reference samples. Figure 5.3 shows the estimated density function at x_{30}, obtained via kernel density estimation from the generated samples.

To quantify performance, we compute the relative error between the one-marginal estimated density of the generated samples and that of the reference samples, averaged across all dimensions. It is worth noting that the target density presents a significant contrast between peaks and wells, making accurate recovery particularly challenging. Despite this, the results show that the mean-field approximated basis achieves a relative error of 0.0322, demonstrating the effectiveness of our proposed approach.

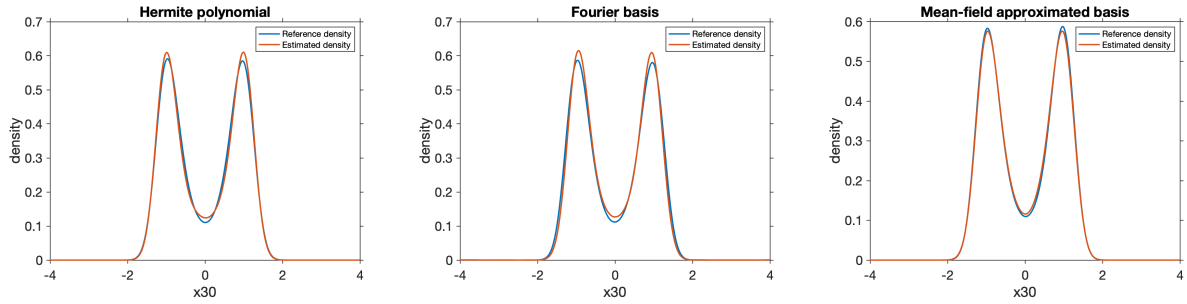


Figure 5.3: 32-dimensional double-well density result. We visualize one-marginal densities at x_{30} for three choices of eigenfunctions (Left) Hermite polynomial, (middle) Fourier basis, and (Right) Mean-field approximated basis. The relative one-marginal density errors are 0.0472, 0.0665, 0.0322, respectively.

5.6.3 High-dimensional Ginzburg-Landau Density

In this subsection, we consider a further complicated density model, called Ginzburg-Landau model:

$$\rho_0 = \frac{1}{Z}\exp(-\beta_{\mathrm{GL}} V_{\mathrm{GL}}(x)), V_{\mathrm{GL}}(x) = \sum_{i=1}^{d}\left(\frac{\lambda_{\mathrm{GL}}}{2}\left(\frac{x_i - x_{i-1}}{h}\right)^2 + \frac{1}{4\lambda_{\mathrm{GL}}}(1-x_i^2)^2\right),$$

for $x \in [-L, L]^d$, where $x_0 = x_d$. In addition, the parameter β_{GL} denotes the inverse temperature, while λ_{GL} and h control the interaction strength between different dimensional coordinates. In the numerical experiments, we fix $\beta_{\mathrm{GL}} = 1/8$ and $h = 0.1$, and consider various choices of λ_{GL} to evaluate the performance of our method under different interaction strength settings.

For all experiments in this subsection, we use $n = 10$ basis functions per dimension. A total of 80,000 samples are drawn from the initial distribution to estimate the score function over the interval $t \in [0, T]$ with $T = 2.0$ and $\Delta t = 0.002$. For the score representation using the local cluster structure (5.20), we set the bandwidth size to $d_{\mathrm{b}} = 4$. We evaluate all three choices of eigenfunctions in our experiments. For the Hermite polynomial basis, we fix $\beta = 1.0$ and $V(x) = \frac{1}{2}\|x\|_2^2$. For the Fourier basis, we set $L = 5$ and $\beta = 0.25$. For

the mean-field approximated basis, we fix $\beta = 1.0$ and construct the approximated density by matching the first 6 moments in the entropy maximization procedure. In addition, for the fast numerical technique described in (5.21) (Section 5.4.3), we select the rank that minimizes the relative error in the initial score estimation and use this same rank for score computation at all subsequent time steps t.

In the reverse-time SDE step, we generate 80,000 samples from the base distribution and evolve them according to the learned dynamics, with time-step $\Delta t = 0.002$. To visualize the structure of the non-mean-field target density, we focus primarily on two-dimensional marginal plots, which highlight the accuracy of interactions between pairs of dimensional coordinates. To quantitatively assess the global accuracy of the generated samples, we compute the second-moment error:

$$\frac{\|\tilde{X}^T\tilde{X} - X^{*T}X^*\|_F}{\|X^{*T}X^*\|_F},$$

where $\tilde{X} \in \mathbb{R}^{N\times d}$ is the matrix of generated samples, and $X^* \in \mathbb{R}^{N\times d}$ is the matrix of reference samples.

32d Result

In this subsection, we present results for the 32-dimensional case, focusing on two choices of the interaction strength parameter. The weak interaction setting uses $\lambda_{\mathrm{GL}} = 0.03$, while the strong interaction setting uses $\lambda_{\mathrm{GL}} = 0.05$. These two choices already exhibit distinct patterns in the density 2-marginal visualizations, as illustrated in Figure 5.4. The relative second-moment errors are reported in Table 5.1, which shows that the mean-field approximated basis achieves the best performance among the tested methods.

Furthermore, Figure 5.4 visualizes the two-dimensional marginal densities obtained using the mean-field approximated basis. The small discrepancy between the generated and

reference distributions illustrates the effectiveness of our approach in capturing complex interactions.

	Mean-field approximated basis	Fourier basis	Hermite polynomial
strong interaction	0.0512	0.0575	0.0588
weak interaction	0.0657	0.0652	0.0685

Table 5.1: 32-dimensional Ginzburg-Landau density result. We include the relative 2nd-moment error for both strong interaction case and weak interaction case among three choices of eigenfunctions.

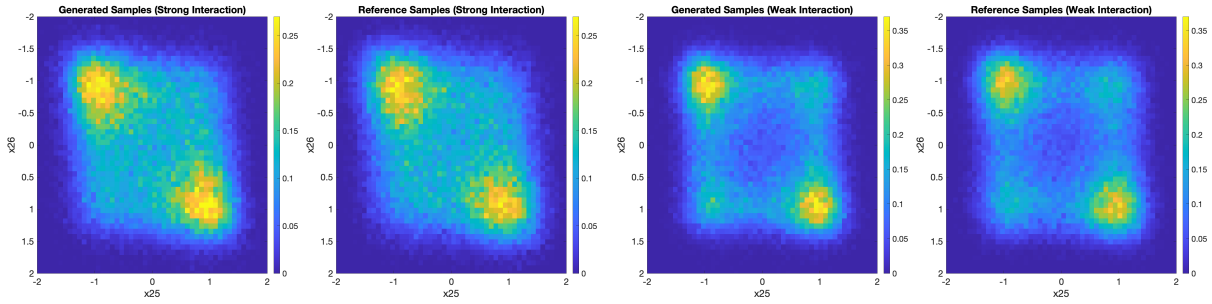


Figure 5.4: 32-dimensional Ginzburg-Landau density result with mean-field approximated basis. We visualize two-marginal densities at (x_{25}, x_{26}) for (Left) strong interaction setting and (right) weak interaction setting.

64d Result

We also evaluate our method in a 64-dimensional setting, keeping the interaction parameter fixed at $\lambda_{\mathrm{GL}} = 0.05$. In this experiment, we primarily use the mean-field approximated basis, while keeping all other parameters identical to those used in the 32-dimensional case.

Figure 5.5 presents selected two-marginal plots comparing the generated samples with the ground truth. The close agreement in second-moment statistics further supports the accuracy and robustness of our approach in high-dimensional settings.

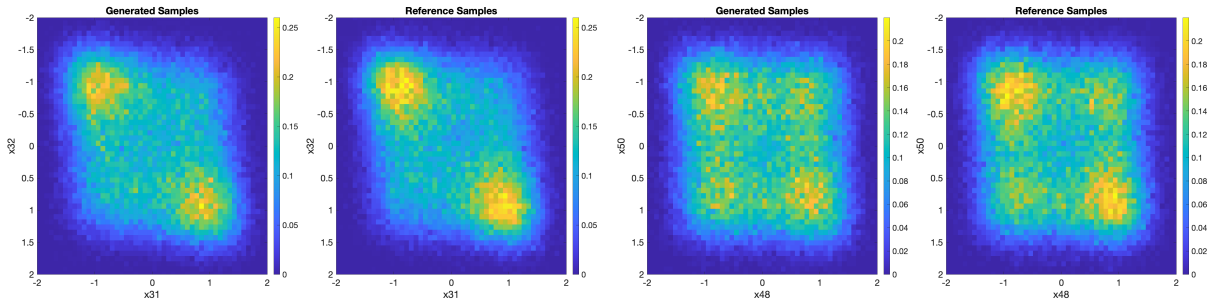


Figure 5.5: 64-dimensional Ginzburg-Landau density result with mean-field approximated basis. We visualize two-marginal densities (left) (x_{31}, x_{32}) and (right) (x_{48}, x_{50}). The relative 2nd moment error is 0.0745.

5.6.4 MNIST Dataset

In the final example, we apply our algorithm to the MNIST dataset, which consists of 70,000 images of handwritten digits (0 through 9) where each image is a 28×28 pixel matrix. In our approach, for each digit, we use 5,000 samples as training data to estimate the score function. Note that the dimensionality of each sample is $d = 28^2 = 784$, which is relatively high. To reduce the computational complexity, we first perform PCA on the training data and retain the first 10 principal components. We then apply our approach in this reduced space to generate new samples in the principal component domain. Finally, we map the generated principal components back to the original image space to recover the digit images.

In the numerical experiments, we use 10 Fourier basis functions per dimension to learn the score function over the interval $t \in [0, 3]$ with time step $\Delta t = 0.002$, with a temperature parameter $\beta = 0.5$ and domain size $L = 4$. The rank used in the fast low-rank approximation technique follows the same selection strategy as described in Section 5.6.3. Figure 5.6 shows both the generated and real MNIST images. The results clearly demonstrate that the images generated by our method successfully preserve the structural features of the digits.

Figure 5.6: MNIST dataset result. (Left) Generated Images. (Right) Real Images. For each digit, we choose five images randomly.

5.7 Conclusion

In this work, we introduced an alternative approach for diffusion models that avoids both iterative optimization and forward SDE. By representing the score function in a sparse set of eigenbasis of the backward Kolmogorov operator associated with a base distribution, we reformulated score estimation as a closed-form linear system problem—eliminating the need for iterative optimization and stochastic simulation. We provided a theoretical analysis of the approximation error using perturbation theory, showing that the score can be accurately

estimated under mild assumptions. Our numerical experiments on high-dimensional Boltzmann distributions and real-world datasets, such as MNIST, demonstrate that the proposed method achieves competitive performance.

As a direction for future work, we aim to extend this approach to more complex models, such as applications in molecular dynamics and scientific computing, where accurate score estimation is particularly important.

CHAPTER 6

TENSOR SPARSE REGRESSION

6.1 Introduction

We consider the nonparametric regression model

$$y = f^*(\boldsymbol{x}) + \epsilon, \tag{6.1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the Gaussian noise. We assume that the regression function f^* admits a sparse expansion over a prescribed set of multivariate candidate features $\{\Phi_k(\boldsymbol{x})\}_{k=1}^p$:

$$f^*(\boldsymbol{x}) = \sum_{k \in \mathcal{S}^*} \Phi_k(\boldsymbol{x})\, \beta_k^*, \tag{6.2}$$

where $\mathcal{S}^* \subset \{1, \ldots, p\}$ is an unknown sparse index set and $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is the sparse coefficient vector. Given i.i.d. observations $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^N$, our goal is to estimate the sparse coefficients $\boldsymbol{\beta}^*$ (and hence recover $\mathcal{S}^*$) from data:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{\beta} \text{ is sparse}} \ \frac{1}{N} \sum_{i=1}^N \Big(y^{(i)} - \sum_{k=1}^p \Phi_k(\boldsymbol{x}^{(i)})\, \beta_k \Big)^2. \tag{6.3}$$

A variety of popular methods have been proposed to solve this problem. A foundational example is the Lasso, which augments the least-squares objective with an ℓ_1 penalty and thereby performs estimation and variable selection simultaneously [212]. A closely related method is least angle regression (LARS), which provides an efficient path-following procedure and can be adapted to recover the full Lasso solution path [62]. To better accommodate strongly correlated predictors, the Elastic Net incorporates both ℓ_1 and ℓ_2 penalties, often yielding improved stability and grouped variable selection compared with the Lasso [246]. These methods are commonly solved by first-order algorithms, particularly coordinate de-

scent methods [64, 236], whose computational cost typically scales with the size of the design matrix and is often expressed as $O(TNp)$, where T is the number of iterations. Although these approaches are effective when the number of features p is moderate, they may become prohibitively expensive in high-dimensional regimes where p increases rapidly with the dimensionality d.

In this work, we consider a structured family of candidate features that is substantially richer than univariate additive models, while avoiding the full combinatorial complexity of unrestricted tensor-product expansions. Specifically, we consider truncated ANOVA decompositions of a multivariate function f up to interaction order K:

$$f(\boldsymbol{x}) = \sum_{j} f_j(x_j) + \sum_{j_1<j_2} f_{j_1,j_2}(x_{j_1}, x_{j_2}) + \cdots + \sum_{j_1<\cdots<j_K} f_{j_1,\ldots,j_K}(x_{j_1}, \ldots, x_{j_K}). \quad (6.4)$$

For the sparse structure, we assume the number of non-zero terms in the expansion is independent on dimensionality. To achieve sparse recovery within this model class, without additional prior knowledge, the number of candidate features necessarily scales as $p = O(d^K)$. Consequently, even if the iteration count T is regarded as a constant, the computational cost of a direct sparse regression procedure is at least $O(TNp) = O(Nd^K)$, which becomes expensive and superlinear in the dimensionality d whenever $K \geq 2$.

The main idea of this paper is to develop an alternative sparse regression methodology that exploits the structure of the expansion in (6.4) and achieves improved computational scaling with respect to the dimensionality d. We summarize our approach and contributions in the following:

1. We first apply kernel ridge regression with a carefully chosen kernel spectrum so that the ridge regularization imposes a stronger structured penalty on coefficients associated with higher-order interaction components, thereby reducing their contribution to the estimator and capturing the structure in (6.4).

2. To achieve sparse coefficient recovery, we plan to identify the relative importance of the coefficients and sample a small subset of dominant terms. Since direct sampling from the full coefficient is computationally expensive, we first compress the coefficient obtained from kernel ridge regression into a tensor-train representation. Both tensor-train compression and sampling can be performed efficiently, with computational complexity linear in the dimensionality.

3. Finally, we incorporate a refinement step by solving a reduced sparse regression problem restricted to the selected features. Numerical experiments show that our approach attains competitive accuracy while requiring substantially lower computational cost than classical sparse solvers, demonstrating its practical advantage in high-dimensional settings.

For the organization, Section 6.2 introduces the proposed sparse tensor regression framework. The kernel ridge regression step is described in Section 6.2.1, the TT compression procedure in Section 6.2.2, and the coefficient-sampling stage in Section 6.2.3. The final step consists of a standard sparse regression routine and is therefore omitted for brevity. Section 6.3 presents the theoretical analysis of the estimation error for the proposed estimator. Section 6.4 reports numerical experiments that demonstrate both the accuracy and the computational efficiency of our approach. Finally, Section 6.5 summarizes the main contributions of this work and discusses directions for future research.

Here are some related work.

Sparse Regression. High-dimensional nonlinear sparse regression concerns the estimation of a nonlinear regression function in settings where the dimensionality is large, under the assumption that the underlying signal depends only on a relatively small subset of covariates. Because fully nonparametric regression in high dimensions is generally infeasible due to the curse of dimensionality, the existing literature has emphasized models that combine nonlinear flexibility with additional low-dimensional structure. Among the most influential frameworks

are sparse additive models, in which the regression function is represented as a sum of univariate component functions and sparsity is imposed on the active components, thereby permitting nonlinear effects while maintaining statistical and computational tractability [182, 151, 242]. Subsequent work has further developed this paradigm through refined estimation procedures, generalized sparse additive formulations, and inferential methods for selected nonlinear effects [107, 87, 75, 40].

Tensor Regression. Tensor regression has emerged as an important framework for modeling relationships involving multidimensional array-valued covariates, particularly in structured high-dimensional settings. A central motivation in this literature is that naive vectorization of tensor data destroys multi-dimensional structure and leads to an excessive number of parameters, whereas low-rank tensor representations provide a natural form of dimension reduction and regularization. A foundational contribution is the work [245], who introduced tensor regression for scalar responses with tensor predictors and exploited a low-rank CP decomposition of the coefficient tensor to obtain scalable estimation and asymptotic theory. This line was subsequently extended by [132] through Tucker tensor regression, which offers greater flexibility by allowing different ranks along different modes of the coefficient tensor. Beyond tensor predictors with scalar responses, the literature has also developed models for tensor-valued responses, including parsimonious tensor response regression [131], as well as tensor-on-tensor regression frameworks [141]. Additional developments include multilinear tensor regression for longitudinal relational and network data [93], Bayesian tensor regression methods for uncertainty quantification and adaptive regularization [77], and more recent extensions incorporating sparsity, generalized responses, quantile regression, and nonlinear additive structure [142, 86, 2]. Compared with the aforementioned literature, we propose a new methodology that leverages tensor-train decomposition and coefficient sampling to recover the sparsity pattern of a high-dimensional regression function.

Tensor-train decomposition and the algorithms. Tensor-train (TT) representations

have become a standard tool in scientific computing for high-dimensional problems in which classical grid-based discretizations are computationally prohibitive, including high-dimensional partial differential equations, parametric and stochastic PDEs, and electronic structure calculations [112, 113, 56]. Beyond numerical analysis, TT parameterizations have also been adopted in machine learning as structured low-rank models for compressing and accelerating large-scale architectures [159]. Their appeal lies in the ability to retain substantial expressive power through moderate TT ranks while achieving storage and computational costs that scale far more favorably than those of unstructured tensor-product representations. Moreover, TT-specific approximation procedures such as TT-SVD and TT-cross enable the construction of accurate low-rank surrogates while requiring access to only a relatively small subset of tensor entries, which is particularly advantageous in large-scale settings where full tensor evaluation is infeasible [168, 166].

6.2 Main Algorithm

A modern strategy for alleviating the computational burden of high-dimensional task is to parameterize multivariate functions using *tensor networks* ([168]), among which the *tensor-train* (TT) format (also known as matrix product state) has emerged as a particularly effective representation. The TT format provides a numerically stable and computationally tractable low-rank factorization for the d-dimensional coefficient tensor induced by tensor-product basis expansions. In our setting, we organize the coefficient vector $\boldsymbol{\beta}$ as a sparse d-dimensional tensor $A \in \mathbb{R}^{n\times\cdots\times n}$, so that the regression function admits the tensor-product expansion

$$f(\boldsymbol{x}) \approx \widehat{f}_{\mathrm{TT}}(\boldsymbol{x}; A) = \sum_{l_1=1}^{n} \cdots \sum_{l_d=1}^{n} A(l_1, \ldots, l_d) \prod_{j=1}^{d} \phi_{l_j}(x_j), \qquad A \in \mathbb{R}^{n\times\cdots\times n}. \tag{6.5}$$

where $\{\phi_l\}_{l=1}^n$ is a collection of univariate basis functions and each $\Phi_k(\boldsymbol{x})$ in (6.2) is the tensor-product basis $\prod_{j=1}^d \phi_{l_j}(x_j)$ in the representation. The TT model assumes that A can be decomposed into a chain of low-rank cores: $G_1 \in \mathbb{R}^{n\times r_1}, G_j \in \mathbb{R}^{r_{j-1}\times n\times r_j}$ $(2 \le j \le d-1), G_d \in \mathbb{R}^{r_{d-1}\times n}$, such that

$$A(l_1,\ldots,l_d) = \sum_{\gamma_1=1}^{r_1} \cdots \sum_{\gamma_{d-1}=1}^{r_{d-1}} G_1(l_1,\gamma_1)\, G_2(\gamma_1,l_2,\gamma_2)\cdots G_d(\gamma_{d-1},l_d). \tag{6.6}$$

When the TT ranks $(r_1,\ldots,r_{d-1})$ are moderate, the number of parameters decreases from n^d to $\sum_{j=1}^d r_{j-1}\, n\, r_j$, $(r_0 = r_d = 1)$ which scales linearly in d. Moreover, evaluation of $\widehat{f}_{\mathrm{TT}}(\boldsymbol{x}; A)$ becomes efficient via contracting the TT cores with the univariate basis function sequentially along the chain.

This approach is motivated by the observation that low-cluster functions can often be well approximated by low-rank tensor-train representations. Consequently, after the smoothing induced by kernel ridge regression, the tensor-train format provides an efficient framework for representing the function class described in (6.4). Furthermore, to achieve sparse recovery of the coefficients, we interpret A as a discrete probability mass function and perform sampling from the tensor-train coefficient tensor for a sufficiently large number of draws. The collection of sampled indices then serves as the recovered sparse support. Such sampling can be carried out efficiently through a chain of conditional probabilities, leading to linear computational complexity with respect to the dimensionality.

In this paper, we therefore develop a structured procedure with the following computationally manageable steps:

1. **Kernel ridge regression**. We consider a reproducing kernel Hilbert space $\mathcal{H}$ and employ kernel ridge regression $\widehat{f}_{\mathrm{KRR}}$ to suppress the influence of coefficients associated with high-order cluster basis functions. In particular, we adopt a specific spectral weighting of the kernel that both attenuates higher-order interactions and enables

efficient evaluation of kernel entries and the resulting regression procedure. Further details are provided in Section 6.2.1.

2. **TT compression of kernel ridge regressor coefficients**. We then compress the coefficient tensor obtained from kernel ridge regression into tensor-train format. In Section 6.2.2, we present an efficient algorithm for this compression step that achieves computational complexity linear in the dimensionality.

$$\widehat{f}_{\mathrm{KRR}}(\boldsymbol{x}) \xrightarrow{\text{TT-compress}} \widehat{f}_{\mathrm{TT}}(\boldsymbol{x}; A) \tag{6.7}$$

3. **Sampling candidate features from the TT coefficient.** Next, we interpret the TT coefficient tensor obtained in previous step as an indicator for the importance of tensor-product features. After a preprocessing step that enforces nonnegativity and normalization, we treat A as a discrete probability mass function over multi-indices $(l_1, \ldots, l_d)$ and draw M samples:

$$\{(l_1^{(k)}, \ldots, l_d^{(k)})\}_{k \in \tilde{\mathcal{S}}} \sim A \;=\; G_1 \circ G_2 \circ \cdots \circ G_d \in \mathbb{R}^{n \times \cdots \times n}, \tag{6.8}$$

where the sampled indices set is denoted as $\tilde{S}$ and $|\tilde{S}| = M$. In Section 6.2.3 we present a TT-based sequential sampling routine whose complexity scales as $O(Mdn)$ under fixed ranks.

4. **Sparse regression on the sampled features.** Finally, we perform sparse regression restricted to the M selected features by solving a reduced sparse problem:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^M, \boldsymbol{\beta} \text{ is sparse}} \; \frac{1}{N} \sum_{i=1}^{N} \Big(y^{(i)} - \sum_{k \in \tilde{S}} \Phi_k(\boldsymbol{x}^{(i)})\, \beta_k \Big)^2 \tag{6.9}$$

The solution $\boldsymbol{\beta}$ of (6.9) constitutes our sparse estimator. This step can be implemented

using standard first-order solvers.

In the following three subsections, we describe the first three steps of the algorithm in detail. The final refitting step is standard and conceptually straightforward, and we therefore omit its description. The last subsection includes the computational complexity discussion.

6.2.1 Kernel Ridge Regression

Kernel ridge regression (KRR) is a nonlinear regression method that combines the flexibility of kernel methods with the stability of ℓ_2 regularization. Let $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) associated with a positive semidefinite kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \Psi(\boldsymbol{x})\mathrm{diag}(\boldsymbol{w})\Psi(\boldsymbol{x}')^\top$. Here $\Psi(\boldsymbol{x}) \in \mathbb{R}^{\infty \times n^d}$ is the concatenation of all combination of d-terms tensor-product multivariate basis (feature map) and $\mathrm{diag}(\boldsymbol{w})$ specifies the user-chosen spectral weights. KRR estimates a function $f \in \mathcal{H}$ by solving

$$\widehat{f}_{\mathrm{KRR}} \;=\; \arg\min_{f \in \mathcal{H}} \; \frac{1}{N} \sum_{i=1}^{N} \big(y^{(i)} - f(\boldsymbol{x}^{(i)})\big)^2 \;+\; \lambda_{\mathrm{KRR}} \|f\|_{\mathcal{H}}^2, \tag{6.10}$$

where $\lambda_{\mathrm{KRR}} > 0$ controls the bias–variance tradeoff and $\|\cdot\|_{\mathcal{H}}$ is the corresponding RKHS norm. By the representer theorem, the minimizer admits the finite expansion $\widehat{f}_{\mathrm{KRR}}(\cdot) = \sum_{i=1}^{N} z_i \, K(\cdot, \boldsymbol{x}^{(i)})$. Substituting the expansion into (6.10) yields the closed-form coefficients

$$\boldsymbol{z} \;=\; (K + N\lambda_{\mathrm{KRR}} I_{N \times N})^{-1} \boldsymbol{y} \in \mathbb{R}^{N \times 1}, \tag{6.11}$$

and the resulting predictor is

$$\widehat{f}_{\mathrm{KRR}}(\cdot) \;=\; K(\cdot, \boldsymbol{x})\boldsymbol{z} = \Psi(\cdot) \left(\mathrm{diag}(\boldsymbol{w})\Psi(\boldsymbol{x})^\top \boldsymbol{z}\right) \tag{6.12}$$

within the bracket, $\mathrm{diag}(\boldsymbol{w})\Psi(\boldsymbol{x})^\top \boldsymbol{z}$ is the coefficient from KRR.

We specify the spectral weighting $\mathrm{diag}(\boldsymbol{w})$ to reflect the interaction order, following the

construction proposed in [176]. Specifically, we assign the weight α^k to each multivariate basis function belonging to the k-cluster set $\mathcal{B}_{k,d}$ defined in (6.13).

$$\mathcal{B}_{k,d} = \left\{ \phi_{l_1}(x_{j_1}) \cdots \phi_{l_k}(x_{j_k}) \;\middle|\; 1 \le j_1 < \cdots < j_k \le d,\ 2 \le l_1, \ldots, l_k \le n \right\}. \tag{6.13}$$

The motivation for this choice is to assign smaller weights for higher-order cluster bases, thereby reducing their influence in the resulting estimator. In addition, under this spectral structure, each entry of the kernel matrix can be evaluated through d tensor-product terms, as illustrated in Figure 6.1, leading to a computational cost of $O(dn)$ per kernel evaluation.

The computational complexity of direct solving inverse in (6.11) is $O(N^3)$, which is prohibitively expensive when N is large. In practice, we can consider Nystörm approximation and choose a subset of rows and columns of kernel matrix to approximate the inverse. This strategy can provide computational cost $O(Nm^2 + m^3)$ which is cheap when the cardinality of subset $m \ll N$. The details are included in [11]. Besides, λ_{KRR} can be selected via cross-validation. By assuming m as a constant, the total computational complexity (including the kernel evaluation) in this step is $O(Ndn)$.

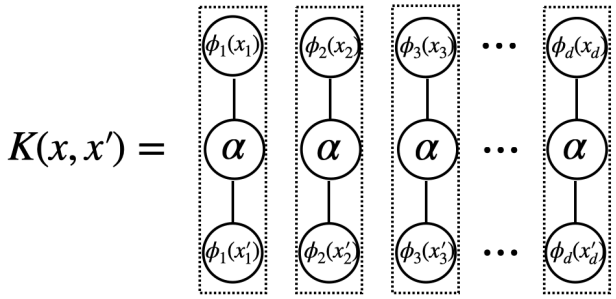


Figure 6.1: Tensor diagram of $K(\boldsymbol{x}, \boldsymbol{x}')$. Middle core α represents $\text{diag}(1, \alpha, \cdots, \alpha)$. The evaluation needs $O(dn)$ cost.

6.2.2 TT Compression

The coefficient obtained from kernel ridge regression already suppresses the contribution of higher-order cluster basis functions. In this subsection, we further show that this coefficient tensor can be efficiently compressed into tensor-train format, which is essential for the efficient sampling procedure developed in the next step.

We use TT-compress$(\cdot)$ to denote the tensor-train compression algorithm. With this notation, the resulting predictor is given by

$$\widehat{f}_{\mathrm{TT}}(\cdot) \;=\; \Psi(\cdot)\left(\text{TT-compress}\left(\mathrm{diag}(\boldsymbol{w})\Psi(\boldsymbol{x})^{\top}\boldsymbol{z}\right)\right) \tag{6.14}$$

With above choice of spectrum $\mathrm{diag}(\boldsymbol{w})$, the input $\mathrm{diag}(\boldsymbol{w})\,\Psi(\boldsymbol{x})^{\top}\boldsymbol{z}$ can be evaluated efficiently by exploiting its tensor-product structure and carrying out the required contractions through tensor-network computations. More precisely, this quantity may be interpreted as a sum of N rank-one tensor-train terms, each of which consists of d tensor-product components. The corresponding computation is illustrated by the tensor diagram in Figure 6.2. For each mode j, we define $\tilde{\Phi}_j(\boldsymbol{x}_j) := \phi_j(\boldsymbol{x}_j)\,\mathrm{diag}(1, \alpha, \cdots, \alpha) \in \mathbb{R}^{n\times 1}$, which collects the evaluations of the weighted univariate basis ϕ_j on the j-th covariate $\boldsymbol{x}_j$. Then this quantity can be completed with computational complexity $O(Ndn)$.

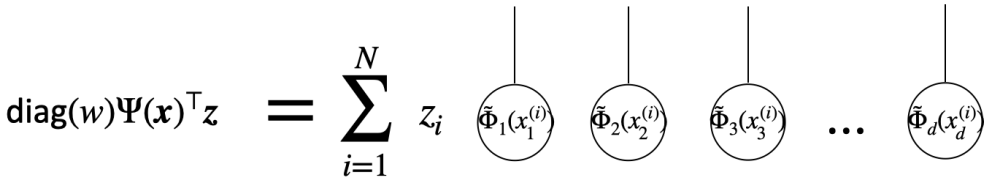


Figure 6.2: Tensor diagram of $\mathrm{diag}(\boldsymbol{w})\,\Psi(\boldsymbol{x})^{\top}\boldsymbol{z}$.

For the TT-compress$(\cdot)$ operator, we adopt the TT-SVD with kernel Nyström acceleration (TT-SVD-kn) developed in Section 4 of previous work [176]. The central idea is to perform a sequence of truncated singular value decompositions (SVDs) on appropriate unfoldings of

the target tensor, thereby constructing a tensor-train representation in a left-to-right sweep. Moreover, we exploit the fact that the tensors arising in our setting admit an N-term rank-one decomposition, so that each truncated SVD can be reformulated in terms of an eigenvalue decomposition of an associated Gram matrix.

More specifically, we define $\hat{c} := \mathrm{diag}(\boldsymbol{w})\,\Psi(\boldsymbol{x})^\top \boldsymbol{z}$, which is viewed as a d-dimensional coefficient tensor. To compute the first core, we reshape coefficient tensor $\hat{c}$ into its first unfolding $\hat{c}^{(1)}$ and consider its Gram matrix $\hat{c}^{(1)}(\hat{c}^{(1)})^\top$. Owing to the structure (sum of rank-one tensor-product) in $\hat{c}^{(1)}$, this computation can be decomposed according to the tensor diagram in Figure 6.3. To unify the notation, we denote $B_1 = \hat{c}^{(1)}$.

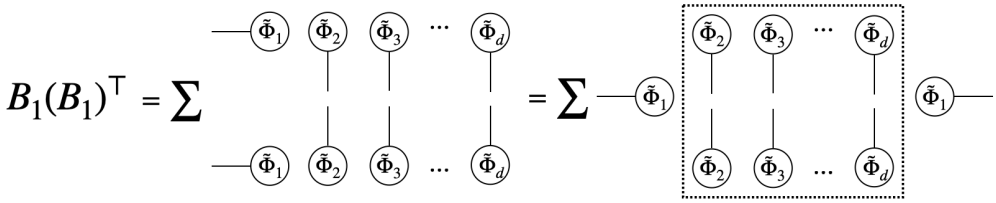


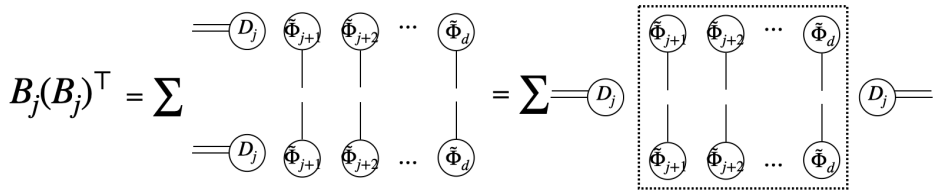
Figure 6.3: Tensor diagram of $B_1 B_1^\top$. For clarity, the weight vector $\boldsymbol{z}$ is omitted from the summation. The Gram matrix is computed by taking the double summation over the three quantities shown in the diagram.

Denoting the middle matrix in the diagram by $E \in \mathbb{R}^{N\times N}$. A direct matrix multiplication for $B_1 B_1^\top$ would require $O(N^2 n^2)$ operations, which is prohibitively expensive when N is large. To accelerate this step, we again employ the Nyström approximation. Letting $\mathcal{I}$ be a subset of row and column indices, we approximate E by

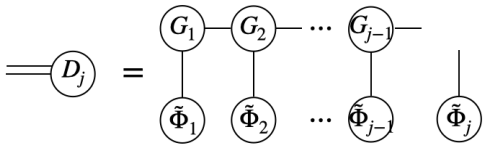
$$E \approx E(:,\mathcal{I})\big(E(\mathcal{I},\mathcal{I})\big)^\dagger E(:,\mathcal{I})^\top. \tag{6.15}$$

When the subset size $|\mathcal{I}|$ is treated as a constant, the cost of this computation is reduced to $O(Nn)$. Further details are provided in [11]. After forming $B_1 B_1^\top$, we compute its eigenvalue decomposition and obtain the first tensor core $G_1 \in \mathbb{R}^{n\times r_1}$ by retaining the top r_1 eigenvectors. Projecting B_1 onto this subspace yields the compressed tensor, from which

the next core is computed through the eigenvalue decomposition of the Gram matrix $B_2 B_2^\top$. The general procedure is illustrated in the Figure 6.4.



(a) Tensor diagram of $B_j B_j^\top$. The Gram matrix is computed by taking the double summation over the three quantities shown in the diagram.



(b) Tensor diagram of D_j. The computation of D_j can be carried out sequentially.

Figure 6.4: Tensor diagrams in Algorithm 11.

Following the same Nyström approximation trick (6.15) for all middle matrix E of Figure 6.4 and the tensor-train contraction operation, each subsequent step can be carried out efficiently. At stage j, we compute the tensor core G_j by selecting the top r_j eigenvectors of the corresponding Gram matrix. Repeating this procedure for all tensor cores yields the full tensor-train decomposition. The total cost of forming all Gram matrices $B_j B_j^\top$ is $O(Ndn)$, and including the cost of the eigenvalue decompositions, the overall complexity of this stage is $O(Ndn + n^3)$. The algorithm is listed in Algorithm 11 (assuming $r_0 = 1$) and we refer to [176] for the complete description and implementation details.

Algorithm 11 TT Compression

INPUT: Tensor $\hat{c} := \mathrm{diag}(\boldsymbol{w})\,\Psi(\boldsymbol{x})^\top \boldsymbol{z}$. Target rank $\{r_1, \cdots, r_{d-1}\}$. Subsampling set $\mathcal{I}$.

1: Reshape coefficient tensor $\hat{c}$ into its first unfolding and denote as B_1.

2: **for** $j \in \{1, \cdots, d-1\}$ **do**

3: Perform eigen-decomposition to $B_j B_j^T \in \mathbb{R}^{r_{j-1}n \times r_{j-1}n}$, whose calculation is visualized in Figure 6.4 with Nystörm acceleration in (6.15) through subsampling set $\mathcal{I}$. Obtain the j-th core $G_j \in \mathbb{R}^{r_{j-1} \times n \times r_j}$, which is reshaped from the first r_j principle eigenvectors.

4: **end for**

5: Obtain the last core: $G_d = B_d \in \mathbb{R}^{r_{d-1} \times n}$.

OUTPUT: Tensor train $A = G_1 \circ G_2 \circ \cdots \circ G_d \in \mathbb{R}^{n \times n \times \cdots \times n}$.

6.2.3 Sampling from TT Coefficient

This section describes how to sample multi-indices from a tensor-train (TT) coefficient tensor A. In general, the entries of A need not be nonnegative, and thus A cannot be directly interpreted as a probability mass function. Since our goal is to extract relative importance information from the coefficients, we apply an entrywise transformation and work with the Hadamard square $A_{\mathrm{sq}} := A \odot A$, which is nonnegative by construction. A key advantage of this choice is that A_{sq} admits an explicit TT representation that can be formed directly from the TT cores of A via Kronecker products of the corresponding TT slices.

Concretely, we define the squared-core tensors $\tilde{G}_j \in \mathbb{R}^{r_{j-1}^2 \times n \times r_j^2}$ by

$$\tilde{G}_j(:, l_j, :) \;=\; G_j(:, l_j, :) \;\otimes\; G_j(:, l_j, :), \qquad l_j = 1, \ldots, n, \quad j = 1, \ldots, d, \tag{6.16}$$

where $\otimes$ denotes the Kronecker product of matrices. Then contracting $\{\tilde{G}_j\}_{j=1}^d$ yields the

entrywise square of A:

$$\begin{aligned} A_{\mathrm{sq}}(l_1,\dots,l_d) &= \tilde{G}_1(l_1,:)\,\tilde{G}_2(:,l_2,:)\cdots\tilde{G}_d(:,l_d) \\ &= \Big(G_1(l_1,:)\cdots G_d(:,l_d)\Big)\ \otimes\ \Big(G_1(l_1,:)\cdots G_d(:,l_d)\Big) \\ &= A(l_1,\dots,l_d)^2, \end{aligned} \tag{6.17}$$

The resulting representation introduces a potential challenge: the TT ranks of A_{sq} are the squares of those of A. Nevertheless, this rank inflation remains manageable when the original TT ranks are sufficiently small.

For notational convenience, we assume throughout this subsection that the TT tensor A has nonnegative entries and is proportional to a discrete probability mass function. Under this assumption, we can draw samples from A using a standard sequential (conditional) sampling scheme.

We decompose A via the chain rule

$$A(l_1,\dots,l_d) = A_1(l_1)\prod_{j=2}^{d} A_j\big(l_j \,\big|\, l_{1:j-1}\big), \tag{6.18}$$

where the TT structure permits efficient evaluation of the conditionals $A_j\big(l_j \,\big|\, l_{1:j-1}\big)$ through successive contractions. Define the left message (a row vector) recursively as

$$L_0^\top = 1, \qquad L_j^\top(l_{1:j}) = L_{j-1}^\top(l_{1:j-1})\,G_j(:,l_j,:) \in \mathbb{R}^{1\times r_j}, \qquad j=1,\dots,d. \tag{6.19}$$

Similarly, define the right normalizer (a column vector) by backward contraction:

$$R_d = 1, \qquad R_{j-1} = \sum_{l_j=1}^{n} G_j(:,l_j,:)\,R_j \in \mathbb{R}^{r_{j-1}\times 1}, \qquad j=d,d-1,\dots,1. \tag{6.20}$$

Then, for each $j \geq 2$, the conditional distribution at step j is

$$A_j\big(l_j \mid l_{1:j-1}\big) = \frac{L_{j-1}^{\top}(l_{1:j-1})\, G_j(:, l_j, :)\, R_j}{L_{j-1}^{\top}(l_{1:j-1})\, R_{j-1}}, \qquad 1 \leq l_j \leq n. \tag{6.21}$$

The marginal distribution for the first coordinate is the special case $j = 1$:

$$A_1(l_1) = \frac{G_1(l_1, :)\, R_1}{R_0}, \qquad R_0 = \sum_{l_1=1}^{n_1} G_1(l_1, :)\, R_1.$$

All the computations are visualized in the tensor diagrams of Figure 6.5.

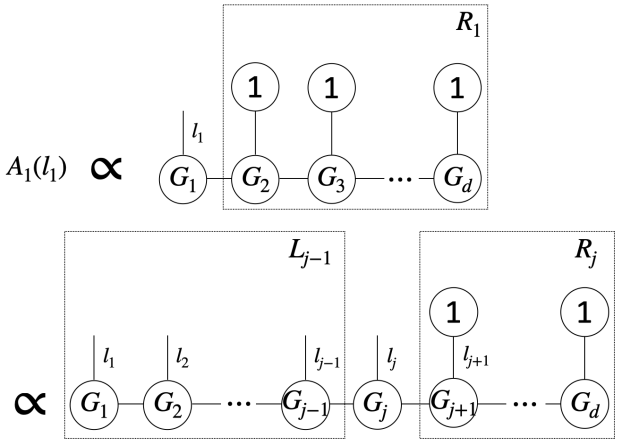


Figure 6.5: Tensor diagram of $A_1(l_1)$ and $A_j\big(l_j \mid l_{1:j-1}\big)$. The core labeled **1** denotes an all-ones vector. For clarity, the normalization constants of the variables are omitted from the diagram.

The algorithm is summarized in Algorithm 12. Concretely, we first precompute the right messages $\{R_j\}_{j=0}^{d}$ via the backward recursion (6.20). We then draw a sample $(l_1, \ldots, l_d)$ sequentially: at step j, we evaluate the conditional probabilities in (6.21), sample l_j accordingly, and update the left message L_j using (6.19). This procedure produces i.i.d. samples from the TT-represented discrete distribution (6.18). The cost per step is $O(r_{j-1}\, n\, r_j)$ to form the full set of conditional discrete distributions $A_j(l_j \mid l_{1:j-1})$ over $l_j \in \{1, \ldots, n\}$, and the overall computational cost is $O\Big(\sum_{j=1}^{d} r_{j-1}\, n\, r_j\Big)$. In particular, when the TT ranks are uniformly bounded, $r_j \leq r$ for all j, this simplifies to $O(Mdnr^2)$, which is commonly

summarized as $O(Mdn)$ up to a rank-dependent constant.

Additionally, there exist alternative methods for performing conditional sampling from the Hadamard square of a tensor-train more efficiently. For example, the recent work [153] combines randomized tensor-train sketching with slice selection via interpolative decomposition to compute the Hadamard product of tensor trains, thereby reducing the computational cost associated with the TT ranks. In this work, we treat the TT ranks as constants and therefore do not explicitly emphasize their contribution to the overall computational complexity.

Algorithm 12 Conditional Sampling from TT Coefficient

INPUT: TT coefficient $A = G_1 \circ \cdots \circ G_d$, number of samples M.

Precompute right normalizer $\{R_j\}_{j=1}^{d}$ in (6.20).
for $k = 1, \ldots, M$ **do**
    Initialize left message $L_0^\top = 1$.
    for $j = 1, \ldots, d$ **do**
        Compute discrete distribution $A_j(l|l_{1:j-1})$ for $1 \le l \le n$ in (6.21) and Figure 6.5.
        Draw sample $\tilde{l}_j^{(k)} \sim A_j(l|l_{1:j-1})$.
        Update L_j in (6.19).
    end for
end for

OUTPUT: Sampled multi-indices $\{(\tilde{l}_1^{(k)}, \ldots, \tilde{l}_d^{(k)})\}_{k=1}^{M}$.

6.2.4 Computational Complexity

In this subsection, we analyze the computational complexity of the proposed method under the assumption that the tensor-train ranks remain bounded by a constant. As shown in Section 6.2.1, the kernel ridge regression step can be carried out with complexity $O(Ndn)$ when combined with the Nyström approximation. In Section 6.2.2, the proposed tensor-train compression procedure also requires only $O(Ndn)$ operations. The sampling stage described in Section 6.2.3 has complexity $O(Mdn)$ when M samples are generated. Finally, in the refitting stage, the iterative solver incurs a cost of $O(TNM)$, where T denotes the number

of iterations. Therefore, the overall computational complexity of the method is

$$O(Ndn + Mdn + TNM). \tag{6.22}$$

The central advantage of the proposed pipeline is the intermediate coefficient-driven sampling step, which constructs a refined candidate set tailored to the data, thereby avoiding optimization over the full K-cluster set of size $p = O(d^K n^K)$ in the classical Lasso estimator. Our analysis and empirical results indicate that accurate support recovery is achieved when the number of samples satisfies $M = O(|\mathcal{S}|)$, where $|\mathcal{S}|$ is the cardinality of the ground truth sparse support. Consequently, when $|\mathcal{S}|$ is independent on d, the total computational cost of the overall procedure scales essentially linearly in the dimensionality: $O(Ndn) + O(|\mathcal{S}|dn) + O(TN|\mathcal{S}|)$, which is markedly smaller than $O(TNd^K n^K)$ required by naive Lasso on the full K-cluster set.

Remark: Our framework also extends naturally to the *single-index* regression model

$$y = f\big(\langle \boldsymbol{x}, \boldsymbol{\beta} \rangle\big) + \epsilon, \tag{6.23}$$

where f is an unknown link function and the index vector $\boldsymbol{\beta} \in \mathbb{R}^d$ is assumed to be sparse. In this setting, the regression function varies primarily along a low-dimensional manifold embedded in the space, suggesting an initial dimension-reduction step. A simple and widely used preprocessing strategy is to apply principal component analysis (PCA) to the covariates $\{\boldsymbol{x}^{(i)}\}_{i=1}^N$ to obtain a low-dimensional representation. One may then estimate the reduced subspace and subsequently apply our TT-based sparse recovery methodology to identify the active coordinates and refine the regression estimate.

6.3 Theoretical Results

In this section, we present the theoretical analysis of the main algorithm. We begin by assuming that the tensor-train estimator provides an accurate approximation to the ground-truth function, in the sense that $\|f^* - \widehat{f}_{\mathrm{TT}}\|_{L_2} \le \varepsilon$. Under this assumption, our primary goal is to analyze the coefficient recovery error introduced by the subsequent sampling and refitting steps in the algorithm. For the simplicity, we choose ordinary least-squares in the refit step. The analysis of the error ε follows from standard arguments, and we defer those details to the end of this section.

Theorem 12. *Suppose the ground-truth regressor is $f^*(\boldsymbol{x}) = \sum_{k\in S^*} \Phi_k(\boldsymbol{x})\beta_k^*$. The basis is orthonormal under distribution $\boldsymbol{x} \sim \mu_x$: $\mathbb{E}_{\mu_x}[\Phi_k(\boldsymbol{x})\Phi_{k'}(\boldsymbol{x})] = \delta_{k,k'}$. Suppose an initial TT estimator $\widehat{f}_{\mathrm{TT}}$ satisfies $\|f^* - \widehat{f}_{\mathrm{TT}}\|_{L_2} \le \varepsilon$. We draw M i.i.d. coefficients of $\widehat{f}_{\mathrm{TT}}$ following Algorithm 12 and let $\widehat{\mathcal{S}}$ be the set of unique sampled indices. Let $\widehat{f}_{\mathrm{sparse}}$ be the ordinary least-squares fit of $\{y^{(i)}\}_{i=1}^N$ on the selected features $\{\Phi_k(\boldsymbol{x}^{(i)})\}_{k\in\widehat{\mathcal{S}}}$.*

Assume a lower bound of ground truth coefficient $m^* := \min_{k\in S^*} |\beta_k^*| > \varepsilon$. Suppose the minimal eigenvalue has a lower bound

$$\lambda_{\min}\left(\frac{1}{N}\tilde{\Psi}^\top\tilde{\Psi}\right) \ge 1-\eta, \qquad \textit{for some } 0<\eta<1$$

where $\tilde{\Psi} = \{\Phi_k(\boldsymbol{x}^{(i)})\}_{k\in\widehat{\mathcal{S}}} \in \mathbb{R}^{N\times|\widehat{\mathcal{S}}|}$. Then

$$\mathbb{E}\big[\|\widehat{\beta}_{\mathrm{sparse}} - \beta^*\|_{L_2}^2\big] \;\le\; \|\beta^*\|_2^2\Big(1+\frac{1}{1-\eta}\Big)\exp\Big(-M\frac{(m^*-\varepsilon)^2}{(\|\beta^*\|_2+\varepsilon)^2}\Big) \;+\; \frac{\sigma^2}{1-\eta}\frac{M}{N}. \quad (6.24)$$

The theorem shows that the relative error of the proposed coefficient estimator, measured against the ground-truth sparse coefficient, decays at an exponential rate as the number of effectively sampled coefficients M increases. At the same time, the contribution of observa-

tional noise introduces an additional variance term whose magnitude grows with the noise level in y. Together, these effects highlight a trade-off and characterize the optimal statistical rate achievable by balancing approximation error (controlled by M) and estimation error due to noise.

Remark. We briefly discuss the estimation error $\|f^* - \widehat{f}_{\mathrm{TT}}\|_{L_2}$. In the first stage, the error of the kernel ridge regression estimator, $\|f^* - \widehat{f}_{\mathrm{KRR}}\|_{L_2}$, can be analyzed using standard results from the literature (For example, Section 7.6 of [11]). For the tensor-train compression step, we note that the sparse ground-truth function f^* admits a tensor-train representation with rank at most $|\mathcal{S}^*|$. Consequently, the error of the tensor-train estimator can be bounded as $\|f^* - \widehat{f}_{\mathrm{TT}}\|_{L_2} \leq \|f^* - \widehat{f}_{\mathrm{KRR}}\|_{L_2} + \|\widehat{f}_{\mathrm{KRR}} - \widehat{f}_{\mathrm{TT}}\|_{L_2} \leq 2\|f^* - \widehat{f}_{\mathrm{KRR}}\|_{L_2}$, where the last inequality follows from the fact that $\widehat{f}_{\mathrm{TT}}$ is the best tensor-train approximation of $\widehat{f}_{\mathrm{KRR}}$ within the prescribed rank class. This property holds, for instance, when $\widehat{f}_{\mathrm{TT}}$ is constructed using the standard TT-SVD algorithm in [168]. Other variant of the compression procedure, including the one introduced in Section 6.2.2, are expected to yield similar guarantees, although establishing such results may require additional technical arguments. Since this analysis is not the primary focus of the present paper, we do not pursue these extensions further here.

6.3.1 Preliminaries

Before proving the main theorem, we first present a useful lemma that plays a key role in the argument.

Lemma 2. *Suppose the ground-truth regressor is* $f^*(x) = \sum_{j\in\mathcal{S}^*} \Phi_j(x)\beta_j^*$ *and we observe* N *i.i.d. samples* $y^{(i)} = f^*(\boldsymbol{x}^{(i)}) + \epsilon^{(i)}$ *with* $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ *where* $\boldsymbol{x}^{(i)} \sim \mu_x$ *and* $\{\Phi_j\}$ *are orthonormal under* μ_x*. Denote the sampled indices set as* $\widehat{\mathcal{S}}$ *and define the approximation error as* $\tau^2 = \sum_{j\in\mathcal{S}^*, j\notin\widehat{\mathcal{S}}}(\beta_j^*)^2$*. Let* $\widehat{f}$ *be the ordinary least-squares fit of* $\{y^{(i)}\}_{i=1}^N$ *on the features* $\{\Phi_j(\boldsymbol{x}^{(i)})\}_{j\in\widehat{\mathcal{S}}}$*.*

Assume moreover that the design is well-conditioned in the sense that for some $\eta \in (0,1)$,

$$\lambda_{\min}\left(\frac{1}{N}X^\top X\right) \geq 1-\eta$$

where $X \in \mathbb{R}^{N\times p}$ is the design matrix with $X_{i,j} = \Phi_j(\boldsymbol{x}^{(i)})$. Then,

$$\mathbb{E}\big[\|\widehat{f} - f^*\|_{L_2}^2\big] \;\leq\; \Big(1+\frac{1}{1-\eta}\Big)\tau^2 \;+\; \frac{1}{1-\eta}\frac{p}{N}\,\sigma^2 \tag{6.25}$$

Proof. Here is the proof of Lemma. Denote $f^*_{\widehat{\mathcal{S}}}(\boldsymbol{x}) \;=\; \sum_{j\in\widehat{\mathcal{S}}, j\in\mathcal{S}^*}\Phi_j(\boldsymbol{x})\beta_j^*$, and define the unexplained residual

$$r(\boldsymbol{x}) \;=\; f^*(\boldsymbol{x}) - f^*_{\widehat{\mathcal{S}}}(\boldsymbol{x}) \;=\; \sum_{j\in\mathcal{S}^*, j\notin\widehat{\mathcal{S}}}\Phi_j(\boldsymbol{x})\beta_j^*, \qquad \text{so that} \qquad \mathbb{E}\|r\|_{L_2}^2 = \tau^2.$$

For any $\widehat{f}$ in the given span,

$$\|\widehat{f} - f^*\|_{L_2}^2 = \|\widehat{f} - f^*_{\widehat{\mathcal{S}}}\|_{L_2}^2 + \mathbb{E}\|r\|_{L_2}^2 = \|\widehat{f} - f^*_{\widehat{\mathcal{S}}}\|_{L_2}^2 + \tau^2. \tag{6.26}$$

Therefore, in order to bound the first term, we have

$$\boldsymbol{y} = X\boldsymbol{\beta}^*_{\widehat{\mathcal{S}}} + \boldsymbol{u}, \qquad u_i = r(\boldsymbol{x}^{(i)}) + \epsilon^{(i)},$$

where $\boldsymbol{\beta}^*_{\widehat{\mathcal{S}}}$ is the restriction of $\boldsymbol{\beta}^*$ to $\widehat{\mathcal{S}}$. Let $\widehat{\boldsymbol{\beta}}_{\widehat{\mathcal{S}}}$ be the ordinary least-squares estimator and define $\Delta = \widehat{\boldsymbol{\beta}}_{\widehat{\mathcal{S}}} - \boldsymbol{\beta}^*_{\widehat{\mathcal{S}}}$. Then $\Delta = (X^\top X)^{-1}X^\top\boldsymbol{u}$.

Since the basis are orthonormal in $L_2(\mu_x)$, we have

$$\|\widehat{f} - f^*_{\widehat{\mathcal{S}}}\|_{L_2}^2 = \|\Delta\|_2^2. \tag{6.27}$$

Thus, it remains to bound $\mathbb{E}\|\Delta\|_2^2$.

Next we study the generalization bound for least squares. Conditioning on X, we have

$$\begin{aligned}\mathbb{E}\Big[\|\Delta\|_2^2 \mid X\Big] &= \mathbb{E}\Big[\boldsymbol{u}^\top X(X^\top X)^{-2}X^\top \boldsymbol{u} \mid X\Big] \\ &= \operatorname{tr}\Big(X(X^\top X)^{-2}X^\top \mathbb{E}[\boldsymbol{u}\boldsymbol{u}^\top \mid X]\Big). \end{aligned} \tag{6.28}$$

Let $\boldsymbol{r}_X = \big(r(\boldsymbol{x}^{(1)}), \ldots, r(\boldsymbol{x}^{(N)})\big)^\top$ and then

$$\mathbb{E}[\boldsymbol{u}\boldsymbol{u}^\top \mid X] = \boldsymbol{r}_X \boldsymbol{r}_X^\top + \Sigma_X,$$

where $\Sigma_X = \operatorname{diag}\big(\operatorname{Var}(\epsilon^{(1)} \mid \boldsymbol{x}^{(1)}), \ldots, \operatorname{Var}(\epsilon^{(N)} \mid \boldsymbol{x}^{(N)})\big)$. By the variance assumption, $\Sigma_X \preceq \sigma^2 I$. Substituting this decomposition into (6.28) gives

$$\begin{aligned}\mathbb{E}\Big[\|\Delta\|_2^2 \mid X\Big] &= \operatorname{tr}\Big(X(X^\top X)^{-2}X^\top \boldsymbol{r}_X \boldsymbol{r}_X^\top\Big) + \operatorname{tr}\Big(X(X^\top X)^{-2}X^\top \Sigma_X\Big) \\ &= \boldsymbol{r}_X^\top X(X^\top X)^{-2}X^\top \boldsymbol{r}_X + \operatorname{tr}\Big(X(X^\top X)^{-2}X^\top \Sigma_X\Big). \end{aligned} \tag{6.29}$$

Now observe that

$$X(X^\top X)^{-2}X^\top \preceq \lambda_{\max}\big((X^\top X)^{-1}\big)\, X(X^\top X)^{-1}X^\top = \lambda_{\max}\big((X^\top X)^{-1}\big)\, P_X,$$

where $P_X = X(X^\top X)^{-1}X^\top$ is the orthogonal projector onto $\operatorname{col}(X)$. Hence,

$$\begin{aligned}\boldsymbol{r}_X^\top X(X^\top X)^{-2}X^\top \boldsymbol{r}_X &\le \lambda_{\max}\big((X^\top X)^{-1}\big)\, \boldsymbol{r}_X^\top P_X \boldsymbol{r}_X \\ &\le \lambda_{\max}\big((X^\top X)^{-1}\big)\, \|\boldsymbol{r}_X\|_2^2, \end{aligned} \tag{6.30}$$

since $P_X \preceq I$. Similarly, using $\Sigma_X \preceq \sigma^2 I$,

$$\begin{aligned}\operatorname{tr}\Big(X(X^\top X)^{-2}X^\top \Sigma_X\Big) &\le \lambda_{\max}\big((X^\top X)^{-1}\big)\operatorname{tr}(P_X\Sigma_X)\\ &\le \lambda_{\max}\big((X^\top X)^{-1}\big)\,\sigma^2\operatorname{tr}(P_X)\\ &= \lambda_{\max}\big((X^\top X)^{-1}\big)\,\sigma^2 p, \end{aligned} \tag{6.31}$$

where we use $\operatorname{tr}(P_X) = \operatorname{rank}(P_X) = p$. Combining (6.29), (6.30), and (6.31), we obtain

$$\mathbb{E}\Big[\|\Delta\|_2^2 \mid X\Big] \le \lambda_{\max}\big((X^\top X)^{-1}\big)\big(\|\boldsymbol{r}_X\|_2^2 + \sigma^2 p\big).$$

Using the conditioning assumption $\lambda_{\min}\Big(\frac{1}{N}X^\top X\Big) \ge 1-\eta$, we have

$$\lambda_{\max}\big((X^\top X)^{-1}\big) = \frac{1}{\lambda_{\min}(X^\top X)} \le \frac{1}{N(1-\eta)}.$$

Therefore,

$$\mathbb{E}\Big[\|\Delta\|_2^2 \mid X\Big] \le \frac{1}{N(1-\eta)}\big(\|\boldsymbol{r}_X\|_2^2 + \sigma^2 p\big).$$

Taking expectation over X yields

$$\begin{aligned}\mathbb{E}\|\Delta\|_2^2 &\le \frac{1}{N(1-\eta)}\Big(\mathbb{E}\|\boldsymbol{r}_X\|_2^2 + \sigma^2 p\Big)\\ &= \frac{1}{N(1-\eta)}\left(\sum_{i=1}^N \mathbb{E}[r(\boldsymbol{x}^{(i)})^2] + \sigma^2 p\right)\\ &= \frac{1}{N(1-\eta)}\Big(N\tau^2 + \sigma^2 p\Big), \end{aligned} \tag{6.32}$$

Hence, we have

$$\mathbb{E}\big[\|\widehat{f} - f^*\|_{L_2}^2\big] = \tau^2 + \mathbb{E}\|\Delta\|_2^2 \le \tau^2 + \frac{1}{1-\eta}\Big(\tau^2 + \frac{p}{N}\sigma^2\Big) = \Big(1 + \frac{1}{1-\eta}\Big)\tau^2 + \frac{1}{1-\eta}\frac{p}{N}\sigma^2.$$

□

6.3.2 Proof of Theorem

Here we go back and give the proof of the theorem.

Proof. We denote $\widehat{f}_{\mathrm{TT}}(\boldsymbol{x}) = \sum_{j=1}^{D} \Phi_j(\boldsymbol{x})\beta_j, P_j = |\beta_j|^2/Z, Z = \sum_{j=1}^{D} |\beta_j|^2$. Assume that the sparse estimator is constructed by drawing M indices independently with replacement according to the distribution $\{P_j\}_{j=1}^{D}$, and then performing least squares on the selected support. Let $\widehat{\mathcal{S}} \subseteq [D]$ denote the resulting selected index set, and let $\widehat{f}_{\mathcal{S}}(\boldsymbol{x}) = \sum_{j\in\widehat{\mathcal{S}}} \Phi_j(\boldsymbol{x})\widehat{\beta}_j$ be the least-squares estimator restricted to $\widehat{\mathcal{S}}$. Since at most M indices are selected, we have $|\widehat{\mathcal{S}}| \leq M$. For any subset $\widehat{\mathcal{S}}$, define the missed support

$$\mathcal{S}^* \setminus \widehat{\mathcal{S}} = \{j \in \mathcal{S}^* : j \notin \widehat{\mathcal{S}}\}.$$

By the oracle bound from the previous lemma, applied to the model restricted to the selected support $\widehat{\mathcal{S}}$, we have

$$\mathbb{E}\Big[\|\widehat{f}_{\mathcal{S}} - f^*\|_{L_2}^2 \,\Big|\, \widehat{\mathcal{S}}\Big] \leq \left(1 + \frac{1}{1-\eta}\right) \sum_{j\in\mathcal{S}^*\setminus\widehat{\mathcal{S}}} |\beta_j^*|^2 + \frac{1}{1-\eta}\frac{M}{N}\sigma^2.$$

Taking expectation over the random support $\widehat{\mathcal{S}}$ yields

$$\begin{aligned}
\mathbb{E}\Big[\|\widehat{f}_{\mathrm{sparse}} - f^*\|_{L_2}^2\Big] &= \mathbb{E}_{\mathcal{S}}\Big[\mathbb{E}\Big[\|\widehat{f}_{\mathcal{S}} - f^*\|_{L_2}^2 \,\Big|\, \widehat{\mathcal{S}}\Big]\Big] \\
&\leq \mathbb{E}_{\mathcal{S}}\left[\left(1 + \frac{1}{1-\eta}\right) \sum_{j\in\mathcal{S}^*\setminus\widehat{\mathcal{S}}} |\beta_j^*|^2 + \frac{1}{1-\eta}\frac{M}{N}\sigma^2\right] \\
&= \left(1 + \frac{1}{1-\eta}\right) \sum_{j\in\mathcal{S}^*} |\beta_j^*|^2\, \mathbb{P}(j \notin \widehat{\mathcal{S}}) + \frac{1}{1-\eta}\frac{M}{N}\sigma^2. \qquad (6.33)
\end{aligned}$$

Now fix $j \in [D]$. Since each draw selects index j with probability P_j independently, the

probability that j is never selected in the M draws is

$$\mathbb{P}(j \notin \widehat{\mathcal{S}}) = (1 - P_j)^M \leq e^{-MP_j}.$$

Substituting this bound into (6.33), we obtain

$$\mathbb{E}\Big[\|\widehat{f}_{\text{sparse}} - f^*\|_{L_2}^2\Big] \leq \left(1 + \frac{1}{1-\eta}\right) \sum_{j \in \mathcal{S}^*} |\beta_j^*|^2 e^{-MP_j} + \frac{1}{1-\eta} \frac{M}{N} \sigma^2. \tag{6.34}$$

It remains to lower bound P_j for $j \in \mathcal{S}^*$. Since the basis $\{\Phi_j\}$ is orthonormal and $\|f^* - \widehat{f}_{\text{TT}}\|_{L_2} \leq \varepsilon$, we have

$$\sum_{j=1}^{D} |\beta_j - \beta_j^*|^2 \leq \varepsilon^2,$$

where we extend $\beta_j^* = 0$ for $j \notin \mathcal{S}^*$. In particular, for each $j \in \mathcal{S}^*$, $|\beta_j| \geq |\beta_j^*| - |\beta_j - \beta_j^*|$. Moreover, $\|\widehat{f}_{\text{TT}}\|_{L_2} \leq \|f^*\|_{L_2} + \|f^* - \widehat{f}_{\text{TT}}\|_{L_2} \leq \|f^*\|_{L_2} + \varepsilon$, and therefore

$$P_j = \frac{|\beta_j|^2}{\|\widehat{f}_{\text{TT}}\|_{L_2}^2} \geq \frac{\big(|\beta_j^*| - |\beta_j - \beta_j^*|\big)^2}{(\|f^*\|_{L_2} + \varepsilon)^2}. \tag{6.35}$$

Furthermore, under the assumption, $m^* = \min_{j \in \mathcal{S}^*} |\beta_j^*| > \varepsilon$, then (6.35) implies that for every $j \in \mathcal{S}^*$,

$$P_j \geq \frac{(m^* - \varepsilon)^2}{(\|f^*\|_{L_2} + \varepsilon)^2}.$$

Substituting this estimate into (6.34), we conclude that

$$\mathbb{E}\Big[\|\widehat{f}_{\text{sparse}} - f^*\|_{L_2}^2\Big] \leq \left(1 + \frac{1}{1-\eta}\right) \|\beta^*\|_2^2 \exp\left(-\frac{M(m^* - \varepsilon)^2}{(\|f^*\|_{L_2} + \varepsilon)^2}\right) + \frac{1}{1-\eta} \frac{M}{N} \sigma^2.$$

Finally, by orthonormality of the basis,

$$\|\widehat{f}_{\text{sparse}} - f^*\|_{L_2}^2 = \|\widehat{\beta}_{\text{sparse}} - \beta^*\|_2^2,$$

which yields the desired result.

□

6.4 Numerical Experiments

In this section, we present numerical experiments comparing several approaches. The first is our proposed *sparse tensor regression* method. The second is *standard tensor-train (TT) regression*, using gradient descent to optimize the tensor-train representation. The third is the *classical lasso* applied to an appropriate feature expansion. Besides, we include the numerical results from neural network as a reference benchmark although this approach cannot capture the sparse property in general.

To highlight the computational advantages of our method, we construct ground-truth regressors as sums of low-order cluster components (e.g., involving only 2-cluster or 3-cluster interactions). In such settings, the lasso baseline typically requires at least $O(Nd^2)$ computational cost to include and process all pairwise (2-cluster) interaction features needed for correct recovery, whereas our approach scales linearly in the dimensionality d under the same experimental design. For simplicity, we draw covariates $\boldsymbol{x}$ independently from the uniform distribution on $[0,1]^d$.

Example 1: The first example considers a sparse clustered function constructed from trigonometric components $f(x) = \sum_{i=1}^{4} \cos(2\pi x_{i+1}) \sin(2\pi x_i)$ and we use Fourier basis in our algorithm. Therefore, the ground-truth function lies exactly in the span of the chosen dictionary and can, in principle, be represented without approximation error. Consequently, this experiment is designed to assess the exact recovery capability of the proposed approach. We use 9 fourier basis as univariate basis in the algorithm and keep sample size $N = 30,000$. As reported in Table 6.1, sparse tensor regression achieves errors on the order of the prescribed algorithmic tolerance across a range of dimensions, closely matching the performance of the classical lasso baseline. In contrast, direct tensor-train regression attains substantially

poorer accuracy, as it lacks an explicit mechanism to identify and exploit the sparse clustered structure of the target function.

	Sparse Tensor Regression	Tensor-train Regression	Classical Lasso
$d = 10$	1.7561e-9	0.0180	9.8551e-9
$d = 20$	3.4236e-9	0.2841	3.1923e-8
$d = 30$	7.4925e-9	0.6601	2.1434e-7
$d = 40$	9.3312e-9	0.8536	4.2324e-7
$d = 50$	9.7823e-9	0.9321	8.2856e-7

Table 6.1: Regression on sparse 2cluster trigonometric function with Fourier basis.

To illustrate the sparse recovery behavior, we consider the case $d = 10$ and visualize all two-cluster coefficients of the tensor-train estimator obtained from Step 2 in Figure 6.6. The plot exhibits four prominent peaks, revealing the sparse structure of the underlying ground-truth function. This provides strong evidence that the subsequent sampling step can effectively identify the important coefficients and sample them with high probability.

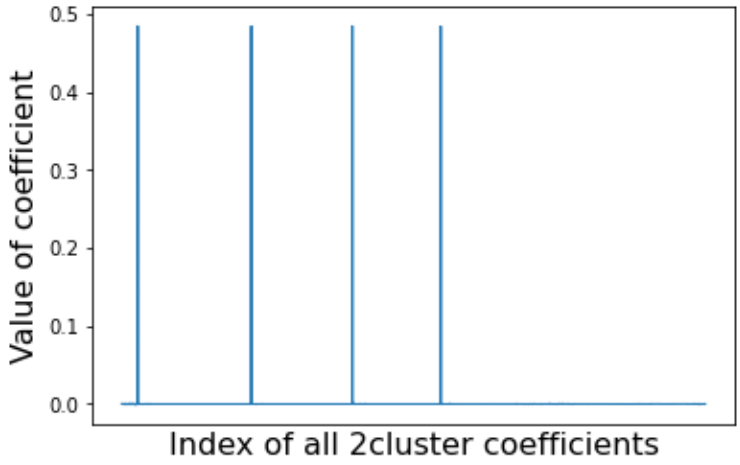


Figure 6.6: Visualization of all two-cluster coefficients of the tensor-train estimator obtained in Step 2 of the algorithm.

Furthermore, we examine performance when a polynomial basis is used. In this setting,

the ground-truth function is generally not represented exactly by the chosen features, and the approximation error is therefore non-negligible. We fix all parameters and Table 6.2 nevertheless shows that our method maintains strong performance across all tested dimensions. This behavior is expected because the target function is sparse and the number of active components does not grow with the dimensionality. In contrast, standard tensor-train regression remains substantially less accurate, since it does not incorporate an explicit sparse recovery step. The neural network baseline performs even worse than the sparse regressor, further illustrating the difficulty of capturing this structured target function without exploiting its intrinsic sparsity.

	Sparse Tensor Regression	Tensor-train Regression	Classical Lasso	NN
$d = 10$	2.2324e-4	0.0128	2.3894e-4	2.2924e-3
$d = 20$	2.2678e-4	0.2297	3.0163e-4	2.7232e-3
$d = 30$	2.3569e-4	0.6116	2.7831e-3	1.8501e-2
$d = 40$	2.2891e-4	0.8201	2.7482e-3	3.5327e-2
$d = 50$	2.2968e-4	0.9238	2.7843e-3	4.9135e-2

Table 6.2: Regression on sparse 2cluster trigonometric function with polynomial basis.

To highlight the computational advantages of our approach, we record the wall-clock runtime of both sparse tensor regression and the classical lasso baseline. As shown in Figure 6.7, the runtime of sparse tensor regression grows approximately linearly with the dimensionality, whereas the lasso runtime increases quadratically due to the need to construct and process all pairwise (2-cluster) basis features. This pronounced difference in scaling behavior underscores the computational benefit of the proposed method in high-dimensional settings.

Example 2: For the second example, we take the ground-truth regressor to be a sparse 2-cluster function formed by a mixture of exponential and sinusoidal components $f(x) = \sum_{i=1}^{3} \sin(2\pi x_i) \exp(x_{i+1})$. We use 9 polynomial basis as univariate basis in the algorithm

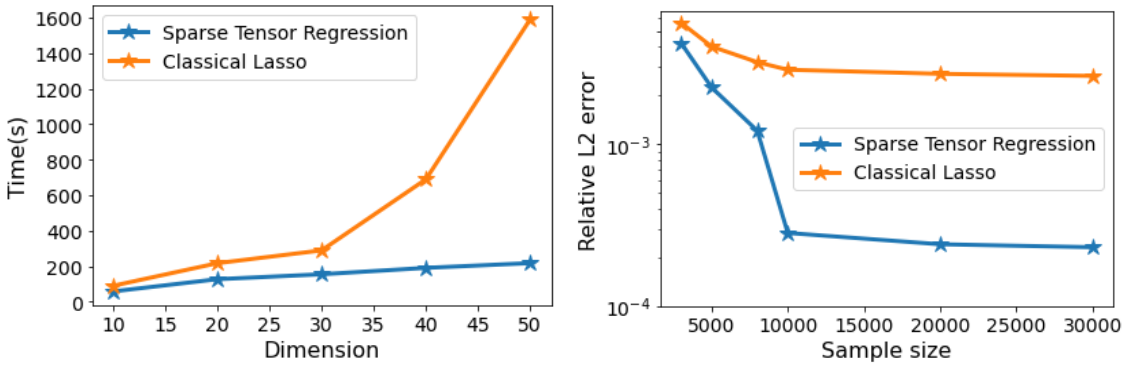


Figure 6.7: Regression on sparse 2cluster trigonometric function with polynomial basis. (Left) The curve of wallclock runtime against dimensionality with fixed $N = 30,000$. (Right) The curve of relative error against sample size with fixed $d = 50$.

and keep the sample size $N = 30,000$. We report results for standard tensor-train regression, the proposed sparse tensor regression, and the classical lasso baseline in Table 6.3. The results show that our method achieves accuracy comparable to that of the lasso sparse regressor. Importantly, it attains this performance with substantially lower computational cost, which constitutes one of the key advantages of the proposed approach.

	Sparse Tensor Regression	Tensor-train Regression	Classical Lasso	NN
$d = 10$	2.2218e-4	0.0039	2.3717e-4	7.2713e-4
$d = 20$	2.2292e-4	0.0658	2.9843e-4	1.4232e-3
$d = 30$	2.2915e-4	0.1286	2.7432e-3	2.4236e-3
$d = 40$	2.3241e-4	0.1613	3.1243e-3	3.2435e-3
$d = 50$	2.4321e-4	0.2058	3.4328e-3	4.6343e-3

Table 6.3: Regression on sparse 2cluster mixed function.

Example3: We consider the following high-dimensional sparse localized spike model on

the domain $[0,1]^d$:

$$f(x) = \sum_{i=1}^{K} \exp\Big(-20(x_i - 0.3)^2 - 20(x_{i+1} - 0.3)^2\Big) + \sum_{i=d-K}^{d} \exp\Big(-20(x_{i-1} - 0.7)^2 - 20(x_i - 0.7)^2\Big). \tag{6.36}$$

This function is composed of a small number of localized Gaussian-type spikes, each depending only on two neighboring variables. The first group of terms generates spikes centered near $(0.3, 0.3)$ on the coordinate pairs (x_i, x_{i+1}) for $i = 1, \ldots, K$, while the second group generates spikes centered near $(0.7, 0.7)$ on the coordinate pairs (x_{i-1}, x_i) for $i = d-K, \ldots, d$. Hence, although the dimension d may be large, the function itself only involves a sparse collection of low-order interactions concentrated on a few coordinate pairs near the beginning and end of the variable chain. This structure makes the example particularly well suited for illustrating the sparse recovery capability of our method. We fix $K = 3$, sample size as $30,000$, and use 15 polynomial basis as univariate basis in the algorithm. As shown in Table 6.4, the proposed approach achieves strong performance and consistently outperforms the standard baseline method.

	Sparse Tensor Regression	Tensor-train Regression	Classical Lasso	NN
$d = 10$	6.7451e-5	0.0314	1.0432e-4	3.5523e-4
$d = 20$	7.0901e-5	0.1967	4.1934e-3	8.2134e-4
$d = 30$	7.3784e-5	0.2654	4.5231e-3	1.4258e-3
$d = 40$	6.9452e-5	0.3181	4.8323e-3	1.8658e-3
$d = 50$	6.8774e-5	0.4395	5.7693e-3	2.8694e-3

Table 6.4: Regression on sparse localized spike model.

6.5 Conclusion

In this work, we developed a computationally efficient framework for sparse tensor regression in high-dimensional settings. The proposed method combines a kernel ridge regressor with tensor-train (TT) compression, a coefficient-sampling stage and a final sparse regression refinement, yielding an estimator that can exploit low-order clustered sparsity while retaining favorable scalability. In particular, when the number of active components in the target regressor does not increase with the dimensionality, the overall computational complexity of our approach scales linearly in the dimensionality, offering a substantial advantage over classical sparse regression. We complemented the algorithmic development with a theoretical analysis that quantifies the estimation accuracy of the proposed procedure. Under suitable identifiability and signal-to-noise assumptions on the underlying coefficients, we established that the estimation error decreases exponentially fast as the number of sampled coefficients increases, thereby providing guidance for the sampling budget required to reach a desired accuracy level. Finally, extensive numerical experiments across multiple synthetic benchmarks demonstrated that our sparse tensor regression method achieves accuracy exceeding that of the classical approach, while consistently requiring significantly lower computational cost.

CHAPTER 7

CONCLUSION

This thesis develops a numerical linear algebra perspective on generative modeling, with the goal of constructing scalable, interpretable, and theoretically grounded alternatives to standard deep-learning-based approaches. The central theme is that many high-dimensional generative modeling problems possess latent low-rank or low-interaction structure, and that such structure can be exploited effectively through tensor networks and structured spectral approximations. From this perspective, the thesis studies both density estimation and diffusion-based generative modeling.

Chapters 2–4 focus on density estimation. In Chapter 2, we introduced a variance-reduction framework that improves the efficiency and stability of tensor-based density estimation relative to standard higher-order singular value decomposition. This chapter shows that appropriate variance-reduction techniques can substantially extend the practical applicability of tensor methods to moderately high-dimensional settings. In Chapter 3, we proposed a tensor-train-based density estimator based on a convolution–deconvolution formulation. The key contribution of this chapter is to overcome the exponential variance growth that arises in empirical density estimation in high dimensions, thereby enabling stable compression and accurate recovery in tensor-train format with linearly computational complexity in dimensionality. In Chapter 4, we extend these ideas to hierarchical tensor-train representations, aiming to model more complex data distributions. Together, these chapters demonstrate that tensor network representations provide an effective framework for high-dimensional density estimation beyond classical nonparametric methods.

Chapters 5 and 6 turn to diffusion-based generative modeling. In Chapter 5, we developed a numerical linear algebra framework for diffusion models that enables optimization-free score estimation. Rather than relying on the expensive training of overparameterized neural networks, this approach exploits structured representations of the score function arising

from the underlying diffusion dynamics. In Chapter 6, we further incorporated sparse regression techniques to recover sparse low-order structure. By combining structured kernel ridge regression, tensor-train compression, and sparse recovery, the proposed method achieves efficient estimation in high dimensions.

Across all chapters, the proposed methods are supported by extensive numerical experiments, which demonstrate their effectiveness in a variety of settings, as well as by theoretical analysis. These results show that numerical linear algebra is not merely a computational tool for accelerating existing methods, but rather a principled framework for designing generative modeling algorithms with favorable structure and interpretability.

Several directions remain for future work. On the density estimation side, it would be valuable to develop more adaptive basis constructions and to extend the tensor-based framework to broader classes of distributions. On the diffusion side, it would be of interest to further investigate structured approximations of score functions beyond the sparse low-order regime and apply to more complicated dataset. More broadly, the results of this thesis suggest that the interplay between generative modeling and numerical linear algebra remains far from fully explored, and that it provides a promising direction for the development of high-dimensional generative methods.

APPENDIX A

APPENDIX OF TENSOR VARIANCE REDUCED SKETCHING

A.1 Preliminaries

Finite-dimensional Tensors

Let B be a tensor of size $m_1 \times m_2 \times \cdots \times m_d$. Let $B(l_1, \ldots, l_k; l_{k+1}, \ldots, l_d)$ be the k-th unfolding matrix of B such that

$$B(l_1, \ldots, l_k; l_{k+1}, \ldots, l_d) = B(l_1, \ldots, l_d). \tag{A.1}$$

Here $B(l_1, \ldots, l_k; l_{k+1}, \ldots, l_d)$ is a matrix in $\prod_{s=1}^{k} m_s \times \prod_{s=k+1}^{d} m_s$ dimensions, and its element in row $(l_1, \ldots, l_k)$ and column $(l_{k+1}, \ldots, l_d)$ equals to $B(l_1, \ldots, l_d)$ as in (A.1). The matrix $B(l_1, \ldots, l_k; l_{k+1}, \ldots, l_d)$ can be obtained from B by the MATLAB function

$$\text{reshape}\Big(B,\ \prod_{s=1}^{k} m_s,\ \prod_{s=k+1}^{d} m_s\Big).$$

Remark 4. *Given $\{l_k\}_{k=1}^{d}$ such that $1 \le \mu_k \le m_k$, we can identify the integer $(l_1, \ldots, l_k)$ in the following recursive way. Set $(\mu_1, \mu_2) = (\mu_1 - 1) \cdot m_2 + \mu_2$. So $1 \le (\mu_1, \mu_2) \le m_1 m_2$. Then $(\mu_1, l_2, l_3) = ((\mu_1, \mu_2), l_3) = ((\mu_1, \mu_2) - 1) \cdot m_3 + l_3$, and so on. When working with $B(l_1, \ldots, l_k; l_{k+1}, \ldots, l_d)$, the k-th unfolding matrix of the tensor B, it suffices to use $(l_1, \ldots, l_k)$ as the row index, and $(l_{k+1}, \ldots, l_d)$ as the column index. Therefore it is not necessary to keep track of the exact value of the integer $(l_1, \ldots, l_k)$.*

Let $Q_k \in \mathbb{R}^{m_k \times q_k}$ be a matrix. Then $B \times_k Q_k$ is a tensor of size $m_1 \times \ldots \times m_{k-1} \times q_k \times m_{k+1} \times \ldots m_d$ such that

$$(B \times_k Q_k)(l_1, \ldots, l_{k-1}, \eta_k, l_{k+1}, \ldots, l_d) = \sum_{l_k=1}^{q_k} B(l_1, \ldots, l_{k-1}, l_k, l_{k+1}, \ldots, l_d) Q_k(l_k, \eta_k)$$

Tensors in Function Spaces

We first introduce a convenient notation for integration of two functions. For $j = 1, 2, 3$ let $\Omega_j \subset \mathbb{R}^{d_j}$ be any regular domain. Let $A(x, y) \in L_2(\Omega_1 \times \Omega_2)$ and $Q(y, z) \in L_2(\Omega_2 \times \Omega_3)$ be two functions. Then

$$(A \times_y Q)(x, z) = \int_{\Omega_2} A(x, y)Q(y, z)dy. \tag{A.2}$$

Let $\mathcal{O} \subset \mathbb{R}$ be a measurable subset. For arbitrary univariate functions $\{f_k(z_k)\}_{k=1}^d$, denote $f_1 \otimes \cdots \otimes f_d \in L_2(\mathcal{O}^d)$ as

$$f_1 \otimes \cdots \otimes f_d(z_1, \ldots, z_d) = f_1(z_1) \cdots f_d(z_d). \tag{A.3}$$

Let $\{\mathcal{S}_j\}_{j=1}^d \subset L_2(\mathcal{O})$ be arbitrary subspaces, where functions in $\mathcal{S}_j$ correspond to the variable z_j. Define

$$\mathcal{S}_1 \otimes \cdots \otimes \mathcal{S}_d = \mathrm{Span}\{g_1(z_1) \cdots g_d(z_d) : g_1(z_1) \in \mathcal{S}_1, \ldots, g_d(z_d) \in \mathcal{S}_d\}.$$

So $\mathcal{S}_1 \otimes \cdots \otimes \mathcal{S}_d \subset L_2(\mathcal{O}^d)$. Let $\{\phi_l\}_{l=1}^\infty$ be a collection of complete basis in $L_2(\mathcal{O})$, and $m \in \mathbb{Z}^+$. For $j \in \{1, \ldots, d\}$, define

$$\mathcal{M}_j = \mathrm{Span}\{\phi_l(z_j)\}_{l=1}^m. \tag{A.4}$$

So $\mathcal{M}_j \subset L_2(\mathcal{O})$ and $\mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d \subset L_2(\mathcal{O}^d)$. Denote

$$\mathcal{P}_{\mathcal{M}_j}(z_j, z_j') = \sum_{l=1}^m \phi_l(z_j)\phi_l(z_j').$$

For any function $A \in L_2(\mathcal{O}^d)$, we define $A \times_1 \mathcal{P}_{\mathcal{M}_1}(z_1, \ldots, z_d)$ using (A.2) as

$$A \times_1 \mathcal{P}_{\mathcal{M}_1}(z_1, \ldots, z_d) = \int_{\mathcal{O}} A(z_1, \ldots, z_{j-1}, z'_j, z_{j+1}, \ldots, z_d) \mathcal{P}_{\mathcal{M}_j}(z'_j, z_j) dz'_j.$$

Therefore $A \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}$ can be defined inductively. A direct calculation shows that

$$A \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}(z_1, \ldots, z_d) = \sum_{l_1=1}^{m} \cdots \sum_{l_d=1}^{m} \langle A, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle \phi_{l_1}(x_1) \cdots \phi_{l_d}(x_d).$$

Therefore $A \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}$ is a function in the space $\mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d$. Given data $\{Z_i\}_{i=1}^N \subset \mathcal{O}^d$, denote the corresponding empirical measure as

$$\widehat{A} = \frac{1}{N} \sum_{i=1}^{N} \delta_{Z_i},$$

where δ_{Z_i} is the indicator function at the point Z_i. We can also project the empirical measure $\widehat{A}$ onto $\mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d$ as

$$\begin{aligned} \widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}(z_1, \ldots, z_d) &= \sum_{l_1=1}^{m} \cdots \sum_{l_d=1}^{m} \langle \widehat{A}, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle \phi_{l_1}(z_1) \cdots \phi_{l_d}(z_d) \\ &= \sum_{l_1=1}^{m} \cdots \sum_{l_d=1}^{m} \left\{ \frac{1}{N} \sum_{i=1}^{N} \phi_{l_1} \otimes \cdots \otimes \phi_{l_d}(Z_i) \right\} \phi_{l_1}(z_1) \cdots \phi_{l_d}(z_d) \end{aligned}$$

Lemma 3. *Suppose $\{Z_i\}_{i=1}^N \subset \mathbb{R}^d$ are sampled independently from the density A^*. Then*

$$\mathbb{E}(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}) = A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d}.$$

Proof. It suffices to observe that $\mathbb{E}(\phi_{l_1} \otimes \cdots \otimes \phi_{l_d}(Z_i)) = \langle A^*, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle$. □

Definition 1 (Coefficient tensors)**.** *Suppose $f \in \mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d \subset L_2(\mathcal{O}^d)$. Let $\mathcal{C}(f)$ be a*

tensor in $\mathbb{R}^{m^d}$ *satisfying*

$$\mathcal{C}(f)(l_1, \ldots, l_d) = \langle f, \phi_{l_1} \otimes \cdots \phi_{l_d} \rangle, \tag{A.5}$$

where $1 \leq l_1, \ldots, l_d \leq m$. *Here,* $\mathcal{C}(f)$ *is the coefficient tensor of* f. *Note that the coefficient tensor* $\mathcal{C}(f)$ *is defined with respect to the basis functions* $\{\phi_l\}_{l=1}^m$.

The following lemma shows that the coefficient tensor is distance preserving, and consequently the space of $\mathcal{M}^{\otimes d}$ and $\mathbb{R}^{m^d}$ have the same topology.

Lemma 4. *Let* $f \in \mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d$, *and let* $\mathcal{C}(f)$ *be defined in* (A.5). *Then*

$$\|f\|_{L_2(\mathcal{O}^d)} = \|\mathcal{C}(f)\|_F.$$

Proof. Let $\{\phi_l\}_{l=1}^\infty$ be the complete basis functions of $L_2(\mathcal{O})$. Then $\{\phi_{l_1} \otimes \cdots \otimes \phi_{l_d}\}_{l_1,\ldots,l_d=1}^\infty$ is a complete basis for $L_2(\mathcal{O}^d)$. Therefore

$$\|f\|_{L_2(\mathcal{O}^d)}^2 = \sum_{l_1=1}^\infty \cdots \sum_{l_d=1}^\infty \langle f, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle^2 = \sum_{l_1=1}^m \cdots \sum_{l_d=1}^m \langle f, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle^2 = \|\mathcal{C}(f)\|_F^2,$$

where the second equality follows from the assumption that $f \in \mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_d$. □

The following lemma demonstrates the compatibility of the coefficient matrices.

Lemma 5. *For* $j = 1, 2, 3$ *let* $\Omega_j \subset \mathbb{R}^{d_j}$ *be any regular domain. Let* $x \in \Omega_1, y \in \Omega_2$, *and* $z \in \Omega_3$ *be three variables in arbitrary dimensions. Let* $\mathcal{M} = \mathrm{Span}\{\phi_l(x)\}_{l=1}^m$, $\mathcal{L} = \mathrm{Span}\{\psi_\eta(y)\}_{\eta=1}^l$, *and* $\mathcal{N} = \mathrm{Span}\{\xi_k(z)\}_{k=1}^n$ *be three subspaces of functions, where* $\{\phi_l(x)\}_{l=1}^m$, $\{\psi_\eta(y)\}_{\eta=1}^l$, *and* $\{\xi_k(z)\}_{k=1}^n$ *are othornomal functions. Let* $A(x, y) \in \mathcal{M} \otimes \mathcal{L}$ *and* $Q(y, z) \in \mathcal{L} \otimes \mathcal{N}$ *be two functions. Then* $A \times_y Q(x, z) = \int_{\Omega_2} A(x, y) Q(y, z) dy$ *satisfies*

$$\mathcal{C}(A \times_y Q) = \mathcal{C}(A)\mathcal{C}(Q),$$

where $\mathcal{C}(A) \in \mathbb{R}^{m\times l}$ *and* $\mathcal{C}(Q) \in \mathbb{R}^{l\times n}$.

Proof. The proof follows from a standard coordinate matching calculation. By definition, the l, k entry of the matrix $\mathcal{C}(A \times_y Q) \in \mathbb{R}^{m\times n}$ is

$$\int_{\Omega_1}\int_{\Omega_3}\int_{\Omega_2} A(x,y)Q(y,z)dy\phi_l(x)\xi_k(z)dzdx.$$

Note that for any function $f(y) \in \mathcal{L}, g(y) \in \mathcal{L}$,

$$\int_{\Omega_2} f(y)g(y)dy = \sum_{\eta=1}^{\infty}\int_{\Omega_2} f(y)\psi_\eta(y)dy\int_{\Omega_2} g(y)\psi_\eta(y)dy = \sum_{\eta=1}^{l}\int_{\Omega_2} f(y)\psi_\eta(y)dy\int_{\Omega_2} g(y)\psi_\eta(y)dy.$$

Since $A(x,y) \in \mathcal{L}$ for fixed x, and $Q(y,z) \in \mathcal{L}$ for fixed z, it follows that

$$\begin{aligned}
&\int_{\Omega_1}\int_{\Omega_3}\int_{\Omega_2} A(x,y)Q(y,z)dy\phi_l(x)\xi_k(z)dzdx\\
=&\int_{\Omega_1}\int_{\Omega_3}\sum_{\eta=1}^{l}\Big\{\int_{\Omega_2} A(x,y)\psi_\eta(y)dy\Big\}\Big\{\int_{\Omega_2} Q(y,z)\psi_\eta(y)dy\Big\}\phi_l(x)\xi_k(z)dxdz\\
=&\sum_{\eta=1}^{l}\Big\{\int_{\Omega_1}\int_{\Omega_2} A(x,y)\psi_\eta(y)\phi_l(x)dydx\Big\}\Big\{\int_{\Omega_2}\int_{\Omega_3} Q(y,z)\psi_\eta(y)\xi_k(z)dydz\Big\}
\end{aligned}$$

which is the (l,k)-entry of the matrix $\mathcal{C}(A)\mathcal{C}(Q)$. □

The coefficient tensor of $A\times_1\mathcal{P}_{\mathcal{M}_1}\times\ldots\times_d\mathcal{P}_{\mathcal{M}_d}$ satisfies

$$\mathcal{C}(A\times_1\mathcal{P}_{\mathcal{M}_1}\times\ldots\times_d\mathcal{P}_{\mathcal{M}_d})(l_1,\ldots,l_d) = \langle A, \phi_{l_1}\otimes\cdots\otimes\phi_{l_d}\rangle. \tag{A.6}$$

For $1 \le k < d$, let $x = (z_1,\ldots,z_k)$ and $y = (z_{k+1},\ldots,z_d)$. Denote

$$\mathcal{M}^{\otimes k} = \mathcal{M}_1\otimes\cdots\otimes\mathcal{M}_k \quad\text{and}\quad \mathcal{M}^{\otimes d-k} = \mathcal{M}_{k+1}\otimes\cdots\otimes\mathcal{M}_d.$$

Then $A\times_x\mathcal{P}_{\mathcal{M}^{\otimes k}}\times_y\mathcal{P}_{\mathcal{M}^{\otimes d-k}}$ is a function in $\mathcal{M}^{\otimes k}\otimes\mathcal{M}^{\otimes d-k}$. The coefficient matrix

$\mathcal{C}(A \times_x \mathcal{P}_{\mathcal{M}^{\otimes k}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-k}}) \in \mathbb{R}^{m^k \times m^{d-k}}$ satisfies

$$\begin{aligned}
&\mathcal{C}(A \times_x \mathcal{P}_{\mathcal{M}^{\otimes k}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-k}})(l_1, \ldots, l_k; l_{k+1}, \ldots, l_d) \\
=&\langle A, (\phi_{l_1} \otimes \cdots \otimes \phi_{l_k}) \otimes (\phi_{l_{k+1}} \otimes \cdots \otimes \phi_{l_d}) \rangle. \\
=&\langle A, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle = \mathcal{C}(A \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d})(l_1, \ldots, l_d).
\end{aligned} \tag{A.7}$$

In other words, $\mathcal{C}(A \times_x \mathcal{P}_{\mathcal{M}^{\otimes k}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-k}})$ is the k-th unfolding matrix of $\mathcal{C}(A \times_1 \mathcal{P}_{\mathcal{M}} \times \ldots \times_d \mathcal{P}_{\mathcal{M}})$.

Best Low Rank Approximations

Throughout the section, we denote $x \in \Omega_1 \subset \mathbb{R}^{d_1}$ and $y \in \Omega_2 \subset \mathbb{R}^{d_2}$. Thus x and y can be variables in arbitrary dimensions.

Theorem 13. *[Singular value decomposition in function space] Let $A(x, y) : \Omega_1 \times \Omega_2 \to \mathbb{R}$ be any function such that $\|A\|_{L_2(\Omega_1 \times \Omega_2)} < \infty$. There exists a collection of singular values $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq 0$, and two collections of orthonormal basis functions $\{\Phi_\rho(x)\}_{\rho=1}^\infty \subset L_2(\Omega_1)$ and $\{\Psi_\rho(y)\}_{\rho=1}^\infty \subset L_2(\Omega_2)$ such that*

$$A(x, y) = \sum_{\rho=1}^{\infty} \sigma_\rho(A) \Phi_\rho(x) \Psi_\rho(y). \tag{A.8}$$

Proof. See Section 6 of [22]. □

Therefore $A(x, y)$ can be viewed as an infinite-dimensional matrix. Suppose in addition that $A(x, y)$ has exactly β many strictly positive singular values, where $\beta \in \mathbb{Z}^+ \cup \{\infty\}$. Then the rank of $A(x, y)$ is β.

Theorem 14. *Let $\{\Phi_\rho(x)\}_{\rho=1}^\infty$ and $\{\Psi_\rho(y)\}_{\rho=1}^\infty$ be defined as in* (A.8)*. For $r \in \mathbb{Z}^+$, denote*

$$[A]_r(x, y) = \sum_{\rho=1}^{r} \sigma_\rho(A) \Phi_\rho(x) \Psi_\rho(y).$$

Then $[A]_r$ is the best low rank approximation of the function A in the sense that

$$\|A(x,y) - [A]_r(x,y)\|_2 = \min_{rank(B)\leq r} \|A(x,y) - B(x,y)\|_2 = \sigma_{r+1}(A(x,y)),$$

$$\|A(x,y) - [A]_r(x,y)\|_{L_2(\Omega_1\times\Omega_2)} = \min_{rank(B)\leq r} \|A(x,y) - B(x,y)\|_{L_2(\Omega_1\times\Omega_2)} = \sqrt{\sum_{k=r+1}^{\infty} \sigma_k^2(A(x,y))}.$$

Proof. See Section 6 of [22]. □

From Theorem 14, it follows that if $B(x,y)$ is a function of rank at most r, then

$$\|A(x,y) - [A]_r(x,y)\|_2 \leq \|A(x,y) - B(x,y)\|_2 \tag{A.9}$$

$$\|A(x,y) - [A]_r(x,y)\|_{L_2(\Omega_1\times\Omega_2)} \leq \|A(x,y) - B(x,y)\|_{L_2(\Omega_1\times\Omega_2)}. \tag{A.10}$$

Definition 2. *Let $A(x,y) : \Omega_1 \times \Omega_2 \to \mathbb{R}$. The range of $A(x,y)$ in the x-variable is defined as*

$$\text{Range}_x(A) = \left\{ f(x) : f(x) = \int A(x,y)g(y)\mathrm{d}y \text{ for any } g(y) \in L_2(\Omega_2) \right\}. \tag{A.11}$$

Suppose $A(x,y) = \sum_{\rho=1}^{\beta} \sigma_\rho(A)\Phi_\rho(x)\Psi_\rho(y)$ as in (A.8). Then an equivalent definition of $\text{Range}_x(A)$ is that

$$\text{Range}_x(A) = \text{Span}\{\Phi_\rho(x)\}_{\rho=1}^{\alpha} = \left\{ \sum_{\rho=1}^{\alpha} a_\rho \Phi_\rho(x) : \{a_\rho\}_{\rho=1}^{\alpha} \subset \mathbb{R} \right\}. \tag{A.12}$$

Consequently, the rank of $A(x,y)$ is the same as the dimensionality of $\text{Range}_x(A)$.

Corollary 2. *Let $x \in \Omega_1, y \in \Omega_2$ be two variables in arbitrary dimensions, and $A(x,y) : \Omega_1 \times \Omega_2 \to \mathbb{R}$ be any function such that $\|A\|_{L_2(\Omega_1\times\Omega_2)} < \infty$. Let $\{\phi_l\}_{l=1}^{\infty}$ and $\{\psi_\eta\}_{\eta=1}^{\infty}$ be two complete basis functions of $L_2(\Omega_1)$ and $L_2(\Omega_2)$ respectively. Let $\mathcal{C}(A) \in \mathbb{R}^{\infty\times\infty}$ be the*

coefficient matrix of A in the sense that

$$\mathcal{C}(A)(l,\eta) = \langle A, \phi_l \otimes \psi_\eta \rangle.$$

(a) $\sigma_\rho(A) = \sigma_\rho(\mathcal{C}(A))$ *for all* $\rho \in \mathbb{Z}^+$. *Here* $\sigma_\rho(A)$ *is the* ρ*-th singular value of* A *when* A *is viewed as a function of two variables* x *and* y.

(b) $\mathcal{C}([A]_r) = [\mathcal{C}(A)]_r$. *Here* $[A]_r$ *is the best rank-r approximation of the function* A*; and* $[\mathcal{C}(A)]_r$ *is the best rank-r approximation of the matrix* $\mathcal{C}(A)$.

Proof. Corollary 2 is a direct consequence of Theorem 14. See Section 6 of [22] for detailed proofs. □

We will use Corollary 2 mainly in the setting where A is a finite rank function. In this case $\mathcal{C}(A)$ can be identify as a finite-dimensional matrix.

Suppose $B(x,y) : \Omega_1 \times \Omega_2$ is a two-variable function with $\Omega_1 \subset \mathbb{R}^{d_1}$ and $\Omega_2 \subset \mathbb{R}^{d_2}$. Suppose in addition that $\|B\|_{L_2(\Omega_1 \times \Omega_2)} < \infty$. Then the operator norm of $B(x,y)$ can be defined as

$$\|B(x,y)\|_2 = \sup_{u \in L_2(\Omega_1), v \in L_2(\Omega_2)} \langle B, u \otimes v \rangle.$$

It is a well known linear algebra result that

$$\|B(x,y)\|_2 \leq \|B\|_{L_2(\Omega_1 \times \Omega_2)}.$$

Whenever the context is clear, for convenience we will write $\|B\|_2$ to indicate the operator norm of $B(x,y)$.

A.2 Proofs Related to Theorem 2

Proof of Theorem 2. By Lemma 11,

$$\|(\widehat{A} - A^*) \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}}\|_2 = \mathrm{O}_{\mathbb{P}}\left(\sqrt{\frac{(m^{d_1} + \ell^{d_2})\log(N)}{N}}\right). \tag{A.13}$$

Suppose this good event holds. Then

$$\begin{aligned}
&\|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_2 \\
\leq &\|A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_2 + \|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}}\|_2 \\
\leq &\|A^* \times_x \mathcal{P}_{\mathcal{M}} - A^*\|_2 \|\mathcal{P}_{\mathcal{S}}\|_2 + \|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}}\|_2 \\
\leq &\,\mathrm{O}_{\mathbb{P}}\left(m^{-\alpha} + \sqrt{\frac{(m^{d_1} + \ell^{d_2})\log(N)}{N}}\right),
\end{aligned} \tag{A.14}$$

where the last inequality follows from Theorem 15, $\|\mathcal{P}_{\mathcal{S}}\|_2 \leq 1$ and (A.13).

By definition, $\mathcal{P}_x^*$ is the projection operator onto $\mathrm{Range}_x(A^*)$ By Lemma 6, $\mathcal{P}_x^*$ is also the projection operator onto $\mathrm{Range}_x(A^* \times_y \mathcal{P}_{\mathcal{S}})$. Therefore by Corollary 5, the Wedin's theorem in Hilbert space, it follows that

$$\|\widehat{\mathcal{P}}_x - \mathcal{P}_x^*\|_2 \leq \frac{\sqrt{2}\|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_2}{\sigma_r(A^* \times_y \mathcal{P}_{\mathcal{S}}) - \sigma_{r+1}(A^* \times_y \mathcal{P}_{\mathcal{S}}) - \|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_2}.$$

By (A.14), we have that $\|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_2 = \mathrm{O}_{\mathbb{P}}\left(m^{-\alpha} + \sqrt{\frac{(m^{d_1} + \ell^{d_2})\log(N)}{N}}\right)$. In addition, since A^* has rank r, it follows that $A^* \times_y \mathcal{P}_S$ has at most rank r. So $\sigma_{r+1}(A^* \times_y \mathcal{P}_{\mathcal{S}}) = 0$. Consequently,

$$\begin{aligned}
&\sigma_r(A^* \times_y \mathcal{P}_{\mathcal{S}}) - \sigma_{r+1}(A^* \times_y \mathcal{P}_{\mathcal{S}}) - \|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_2 \\
&\qquad \geq \sigma_r(A^*)/2 - \|\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{L}} - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_2 \geq \sigma_r(A^*)/4
\end{aligned}$$

where the first inequality follows from Lemma 6, and the last inequality follows from (2.18) in Theorem 2 and (A.14). Therefore

$$\|\widehat{\mathcal{P}}_x - \mathcal{P}_x^*\|_2^2 = \mathrm{O}_{\mathbb{P}}\left\{\frac{1}{\sigma_r^2}\left(m^{-2\alpha} + \frac{(m^{d_1} + \ell^{d_2})\log(N)}{N}\right)\right\}. \tag{A.15}$$

Since $\widehat{\mathcal{P}}_x$ and $\mathcal{P}_x^*$ are both rank r, it follows that $\widehat{\mathcal{P}}_x - \mathcal{P}_x^*$ is at most rank $2r$. Therefore $\|\widehat{\mathcal{P}}_x - \mathcal{P}_x^*\|_{L_2} \leq \sqrt{2r}\|\widehat{\mathcal{P}}_x - \mathcal{P}_x^*\|_2$. This immediately leads to the desired result. □

Lemma 6. *Suppose all the conditions in Theorem 2 hold. Then*

$$\mathrm{Range}_x(A^* \times_y \mathcal{P}_{\mathcal{S}}) = \mathrm{Range}_x(A^*),$$

and that $\sigma_r(A^* \times_y \mathcal{P}_{\mathcal{L}}) > \sigma_r(A^*)/2$.

Proof of Lemma 6. By Lemma 23 in Section A.4 and Theorem 15, the singular values $\{\sigma_\rho(A^* \times_y \mathcal{P}_{\mathcal{S}})\}_{\rho=1}^{\infty}$ of the function $A^* \times_y \mathcal{P}_{\mathcal{S}}(x, y)$ satisfies

$$|\sigma_\rho(A^*) - \sigma_\rho(A^* \times_y \mathcal{P}_{\mathcal{S}})| \leq \|A^* - A^* \times_y \mathcal{P}_{\mathcal{S}}\|_{L_2(\Omega_1 \times \Omega_2)} = \mathrm{O}(\ell^{-\alpha}) \text{ for all } 1 \leq \rho < \infty.$$

As a result if $\sigma_r(A^*) > C_L \ell^{-\alpha}$ for sufficiently large constant C_L, then

$$\sigma_1(A^* \times_y \mathcal{P}_{\mathcal{L}}) \geq \ldots \geq \sigma_r(A^* \times_y \mathcal{P}_{\mathcal{L}}) \geq \sigma_r(A^*) - \|A^* - A^* \times_y \mathcal{P}_{\mathcal{L}}\|_{L_2(\Omega_1 \times \Omega_2)} > \sigma_r(A^*)/2. \tag{A.16}$$

Therefore, the leading r singular values of $A^* \times_y \mathcal{P}_{\mathcal{L}}$ is positive. So the rank of $A^* \times_y \mathcal{P}_{\mathcal{L}}$ is at least r, and therefore the dimensionality of $\mathrm{Range}_x(A^* \times_y \mathcal{P}_{\mathcal{L}})$ is at least r.

Since by construction, $\mathrm{Range}_x(A^* \times_y \mathcal{P}_{\mathcal{L}}) \subset \mathrm{Range}_x(A^*)$. Since in Theorem 2 we assume that the dimensionality of $\mathrm{Range}_x(A^*)$ is r, it follows that $\mathrm{Range}_x(A^* \times_y \mathcal{P}_{\mathcal{L}}) = \mathrm{Range}_x(A^*)$.

□

A.3 Proofs Related to Theorem 3

Assumption 8. *Let $\{\phi_l\}_{l=1}^{\infty}$ be a collection of basis functions in $L_2(\mathcal{O})$ generated from either the reproducing kernel Hilbert spaces or the Legendre polynomials system.*

For $j \in \{1, \ldots, d\}$, let $m \in \mathbb{N}, \ell_j \in \mathbb{N}$, $z_j \in \mathbb{R}$ and denote

$$\mathcal{M}_j = \operatorname{span}\left\{\phi_l(z_j)\right\}_{l=1}^{m} \quad \text{and} \tag{A.17}$$

$$\mathcal{S}_j = \operatorname{span}\left\{\phi_{\eta_1}(z_1)\cdots\phi_{\eta_{j-1}}(z_{j-1})\phi_{\eta_{j+1}}(z_{j+1})\cdots\phi_{\eta_d}(z_d)\right\}_{\eta_1,\ldots,\eta_{j-1},\eta_{j+1},\ldots,\eta_d=1}^{\ell_j}. \tag{A.18}$$

Assumption 9. *Suppose that the data $\{Z_i\}_{i=1}^{N} \subset \mathcal{O}^d$ are independently sampled from the density $A^* : \mathcal{O}^d \to \mathbb{R}^+$. Suppose in addition that $\|A^*\|_{W_2^{\alpha}(\mathcal{O}^d)} < \infty$ with $\alpha \geq 1$ and that $\|A^*\|_{\infty} < \infty$.*

For $j = 1, \ldots, d$, denote $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$, and

$$\sigma_{j,\rho}^* = \sigma_\rho(A^*(z_j, y)), \tag{A.19}$$

where $\sigma_\rho(A^*(z_j, y))$ denote the ρ-th singular value of the two-variable function $A^*(z_j, y)$.

Bias from Projection

Theorem 15. *For $j \in \{1, \ldots, d\}$, let $\mathcal{M}_j$ and $\mathcal{S}_j$ be defined as in* (A.17) *and* (A.18) *respectively. Suppose that Assumption 8 and Assumption 9 hold. Denote $x = z_j$ and $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$. Then*

$$\|A^* - A^* \times_y \mathcal{P}_{\mathcal{S}_j}\|_{L_2(\mathcal{O}^d)}^2 = \mathrm{O}(\ell^{-2\alpha}), \quad \textit{and} \tag{A.20}$$

$$\|A^* - A^* \times_x \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j}\|_{L_2(\mathcal{O}^d)}^2 = \mathrm{O}(m^{-2\alpha} + \ell^{-2\alpha}) \tag{A.21}$$

Proof. The proof of Theorem 15 is detailed in Section A.5. □

Lemma 7. *Let $x = z_j$ and $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$. Then there exists an absolute constant C such that*

$$\sigma_r(\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})) \geq \sigma^*_{j,r} - Cm^{-\alpha}.$$

Here, $\sigma^*_{j,\rho}$ is defined in (A.19) *.*

Proof. By symmetry, it suffices to consider $x = z_1$ and $y = (z_2, \ldots, z_d)$. For brevity, denote

$$\mathcal{M} = \mathcal{M}_1 \quad \text{and} \quad \mathcal{M}^{\otimes d-1} = \mathcal{M}_2 \otimes \cdots \otimes \mathcal{M}_d.$$

Note that $\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}) \in \mathbb{R}^{m \times m^{d-1}}$. By Corollary 2, $\sigma_r(\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})) = \sigma_r(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})$. Therefore

$$\begin{aligned} |\sigma_r(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}) - \sigma_r(A^*(x,y))| &\leq \|A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} - A^*\|_2^2 \\ &\leq \|A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} - A^*\|^2_{L_2(\mathcal{O}^d)} = \mathrm{O}(m^{-2\alpha}), \end{aligned}$$

where the first inequality follows from Lemma 23, and the equality follows from (A.21) with $\mathcal{S}_j = \mathcal{M}^{\otimes d-1}$. The desired result follows from the definition in (A.19) that $\sigma_r(A^*(x,y)) = \sigma^*_{1,r}$. □

Consistency of VRS estimator

For analysis convenient, we can rewrite our main algorithm into Algorithm 13.

Algorithm 13 Multivariable Density Estimation via Variance-Reduced Sketching

INPUT: Data $\{Z_i\}_{i=1}^N$, parameters $\{r_j\}_{j=1}^d \subset \mathbb{Z}^+$, subspaces $\mathcal{M}_j$ as in (A.17) and subspaces $\{\mathcal{S}_j\}_{j=1}^d$ as in (A.18).

Set empirical measures $\widehat{A} = \frac{1}{N/2}\sum_{i=1}^{N/2} \delta_{Z_i}$, and $\widehat{A}' = \frac{1}{N/2}\sum_{i=N/2+1}^{N} \delta_{Z_i}$

[Subspaces estimation via Sketching]

for $j \in \{1, \ldots, d\}$ **do**

Set $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$. Compute

$$\mathcal{C}(\widehat{A} \times_j \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j}) \in \mathbb{R}^{m \times \ell_j^{d-1}}. \tag{A.22}$$

Use SVD to compute $\widehat{U}_j \in \mathbb{O}^{m \times r_j}$, where columns of $\widehat{U}_j$ are the leading r_j left singular vectors of $\mathcal{C}(\widehat{A} \times_j \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j})$. Set $\mathcal{P}_j^U = \widehat{U}_j \widehat{U}_j^\top \in \mathbb{R}^{m \times m}$.

end for

[Sketching using estimated subspaces]

Set $\widehat{\mathcal{C}} = \mathcal{C}(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}) \in \mathbb{R}^{m^d}$.

for $j \in \{1, \ldots, d\}$ **do**

Compute $\widehat{\mathcal{B}}_j = \widehat{\mathcal{C}} \times_1 \mathcal{P}_1^U \times \cdots \times_{j-1} \mathcal{P}_{j-1}^U \times_{j+1} \mathcal{P}_{j+1}^U \times \cdots \times_d \mathcal{P}_d^U \in \mathbb{R}^{m^d}$.

Use SVD to compute $\widehat{V}_j \in \mathbb{O}^{m \times r_j}$, where the columns of $\widehat{V}_j$ are the leading r_j left singular vectors of the matrix $\widehat{\mathcal{B}}_j(l_j; l_1, \ldots, l_{j-1}, l_{j+1}, \ldots, l_d) \in \mathbb{R}^{m \times m^{d-1}}$. Set $\mathcal{P}_j^V = \widehat{V}_j \widehat{V}_j^\top \in \mathbb{R}^{m \times m}$.

end for

Set $\widehat{\mathcal{C}}' = \mathcal{C}(\widehat{A}' \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}) \in \mathbb{R}^{m^d}$. Compute the tensor $\widetilde{\mathcal{C}} = \widehat{\mathcal{C}}' \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V \in \mathbb{R}^{m^d}$.

Compute the estimated density function

$$\widetilde{A}(z_1, \ldots, z_d) = \sum_{l_1=1}^{m_1} \cdots \sum_{l_d=1}^{m_d} \widetilde{\mathcal{C}}_{l_1,\ldots,l_d} \phi_{1,\rho_1}(z_1) \cdots \phi_{d,\rho_d}(z_d).$$

OUTPUT: The estimated density function $\widetilde{A}(z_1, \ldots, z_d)$

Remark 5. *We show that Algorithm 2 and Algorithm 13 are numerically identical. In other words, Algorithm 2 and Algorithm 13 are different interpretations of the same algorithm.*

Note that $\mathcal{P}_j^{\Phi}(z_j, z_j')$ *corresponds to the leading* r_j *singular functions in the variable* z_j *of* $\widehat{A} \times_{z_j} \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j}(z_j, y)$*. In addition,* $\mathcal{P}_j^U$ *corresponds to the leading* r_j *left singular vectors of* $\mathcal{C}(\widehat{A} \times_{z_j} \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j})$*. Therefore* $\mathcal{P}_j^U$ *is the coefficient matrix of* $\mathcal{P}_j^{\Phi}(z_j, z_j')$ *with respect to the basis* $\{\phi_l\}_{l=1}^m$*. In other words*

$$\mathcal{C}(\mathcal{P}_j^{\Phi}) = \mathcal{P}_j^U. \tag{A.23}$$

Observe that the range of $\mathcal{P}_{\mathcal{M}_j}$ *contains* $\mathcal{P}_j^{\Phi}$*, so*

$$\mathcal{P}_{\mathcal{M}_j} \mathcal{P}_j^{\Phi} = \mathcal{P}_j^{\Phi}.$$

Note that $\mathcal{P}_j^{\Psi}(z_j, z_j')$ *corresponds to the leading* r_j *singular functions in the variable* z_j *of*

$$\begin{aligned}
&\widehat{A} \times_j \mathcal{P}_{\mathcal{M}_j} \times_1 \mathcal{P}_1^{\Phi} \times \cdots \times_{j-1} \mathcal{P}_{j-1}^{\Phi} \times_{j+1} \mathcal{P}_{j+1}^{\Phi} \cdots \times_d \mathcal{P}_d^{\Phi} \\
=&\widehat{A} \times_j \mathcal{P}_{\mathcal{M}_j} \times_1 \mathcal{P}_{\mathcal{M}_1} \mathcal{P}_1^{\Phi} \times \cdots \times_{j-1} \mathcal{P}_{\mathcal{M}_{j-1}} \mathcal{P}_{j-1}^{\Phi} \times_{j+1} \mathcal{P}_{\mathcal{M}_{j+1}} \mathcal{P}_{j+1}^{\Phi} \cdots \times_d \mathcal{P}_{\mathcal{M}_d} \mathcal{P}_d^{\Phi} \\
=&\big(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}\big) \times_1 \mathcal{P}_1^{\Phi} \times \cdots \times_{j-1} \mathcal{P}_{j-1}^{\Phi} \times_{j+1} \mathcal{P}_{j+1}^{\Phi} \cdots \times_d \mathcal{P}_d^{\Phi}.
\end{aligned} \tag{A.24}$$

As a result, by (A.23) *and* (A.24)*,*

$$\begin{aligned}
&\mathcal{C}(\widehat{A} \times_j \mathcal{P}_{\mathcal{M}_j} \times_1 \mathcal{P}_1^{\Phi} \times \cdots \times_{j-1} \mathcal{P}_{j-1}^{\Phi} \times_{j+1} \mathcal{P}_{j+1}^{\Phi} \cdots \times_d \mathcal{P}_d^{\Phi}) \\
=&\mathcal{C}\big(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}\big) \times_1 \mathcal{C}(\mathcal{P}_1^{\Phi}) \times \cdots \times_{j-1} \mathcal{C}(\mathcal{P}_{j-1}^{\Phi}) \times_{j+1} \mathcal{C}(\mathcal{P}_{j+1}^{\Phi}) \cdots \times_d \mathcal{C}(\mathcal{P}_d^{\Phi}) \\
=&\mathcal{C}(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}) \times_1 \mathcal{P}_1^U \times \cdots \times_{j-1} \mathcal{P}_{j-1}^U \times_{j+1} \mathcal{P}_{j+1}^U \cdots \times_d \mathcal{P}_d^U.
\end{aligned}$$

So $\widehat{\mathcal{B}}_j$ is the coefficient matrix of $\widehat{B}_j$, where $\widehat{B}_j$ is defined in Algorithm 2 and $\widehat{\mathcal{B}}_j$ is defined in Algorithm 13. Consequently, $\mathcal{P}_j^V$ is the coefficient matrix of $\mathcal{P}_j^\Psi(z_j, z_j')$ with respect to the basis $\{\phi_l\}_{l=1}^m$. In other words

$$\mathcal{C}(\mathcal{P}_j^\Psi) = \mathcal{P}_j^V.$$

Therefore $\widetilde{\mathcal{C}}$ in Algorithm 13 is the coefficient tensor of $\widetilde{A}$ in Algorithm 2, and thus Algorithm 2 and Algorithm 13 are numerically identical

Definition 3 (Range and rank of multivariable functions)**.** *Let $B(z_1, \ldots, z_d) \in L_2(\mathcal{O}^d)$. For $j \in \{1, \ldots, d\}$, let $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$. Define*

$$\mathrm{Range}_j(B) = \left\{ \int_{O^{d-1}} B(z_j, y) g(y) dy : \ g(y) \text{ is any function in } L_2(\mathcal{O}^{d-1}) \right\}.$$

Therefore $\mathrm{Range}_j(B)$ *is a linear subspace in $L_2(\mathcal{O})$. In addition, define* $\mathrm{Rank}_j(B)$ *the dimensionality of* $\mathrm{Range}_j(B)$.

Denote

$$\xi^*_{(r_1,\ldots,r_d)} = \min_{B \in L_2(\mathcal{O}^d): \mathrm{Rank}_j(B) \le r_j} \{\|A^* - B\|_{L_2(\mathcal{O}^d)}\}. \tag{A.25}$$

In other words, $\xi^*_{(r_1,\ldots,r_d)}$ is the error of the best rank-$(r_1, \ldots, r_d)$ approximation of the density A^* in $L_2(\mathcal{O}^d)$.

Proof of Theorem 3. Let $\widetilde{\mathcal{C}} = \widehat{\mathcal{C}}' \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V \in \mathbb{R}^{m^d}$ be as in Algorithm 13. It follows that

$$\mathcal{C}(\widetilde{A}) = \widetilde{\mathcal{C}},$$

where $\mathcal{C}(\widetilde{A})$ denotes the coefficient tensor of $\widetilde{A}$ corresponding to the basis functions $\{\phi_l\}_{l=1}^\infty$ as in Definition 1. Then

$$\|\widetilde{A} - A^*\|_{L_2(\mathcal{O}^d)} \le \|\widetilde{A} - A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}\|_{L_2(\mathcal{O}^d)} + \|A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d} - A^*\|_{L_2(\mathcal{O}^d)}.$$

Note that with high probability,

$$\begin{aligned}\|\widetilde{A} - A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}\|_{L_2(\mathcal{O}^d)} &= \|\widetilde{\mathcal{C}} - \mathcal{C}(A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})\|_F \\ &\leq C_1\left(\sqrt{\frac{mr_1r_2\cdots r_d}{N}} + \xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}\right),\end{aligned}$$

where the equality follows from Lemma 4, and the inequality follows from Lemma 8. In addition by Theorem 15, there exists an absolute constant C_2 such that

$$\|A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d} - A^*\|_{L_2(\mathcal{O}^d)} \leq C_2 m^{-\alpha}.$$

Therefore, with high probability,

$$\|\widetilde{A} - A^*\|_{L_2(\mathcal{O}^d)} \leq C_3\left(\sqrt{\frac{mr_1r_2\cdots r_d}{N}} + \xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}\right).$$

The desired result follows from the fact that $m \asymp N^{1/(2\alpha+1)}$. □

Lemma 8. *Suppose all the conditions as in Theorem 3 hold. Let*

$$\widehat{\mathcal{C}}' = \mathcal{C}(\widehat{A}' \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}) \in \mathbb{R}^{m^d}$$

as in the algorithm, and $\widetilde{\mathcal{C}} = \widehat{\mathcal{C}}' \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V \in \mathbb{R}^{m^d}$ *as in Algorithm 13. Then with high probability,*

$$\|\widetilde{\mathcal{C}} - \mathcal{C}(A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})\|_F \leq C\left(\sqrt{\frac{mr_1r_2\cdots r_d}{N}} + \xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}\right),$$

Here C is some absolute constant independent of N, m, and $\{r_j\}_{j=1}^d$, and $\mathcal{C}(A^*)$ is the coefficient tensor of A^* corresponding to the basis $\{\phi_k\}_{k=1}^m$. The definition of coefficient tensor can be found in Definition 1.

Proof. For convenience, denote $\mathcal{C}^* = \mathcal{C}(A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}) \in \mathbb{R}^{m^d}$. Note that

$$\begin{aligned}
&\|\widetilde{\mathcal{C}} - \mathcal{C}^*\|_F = \|\widehat{\mathcal{C}'} \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V - \mathcal{C}^*\|_F \\
\leq &\|\mathcal{C}^* - \mathcal{C}^* \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F + \|(\mathcal{C}^* - \widehat{\mathcal{C}'}) \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F. \qquad \text{(A.26)}
\end{aligned}$$

We bound $\|\mathcal{C}^* - \mathcal{C}^* \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F$ in **Step 1** to **Step 4**, and bound $\|(\mathcal{C}^* - \widehat{\mathcal{C}'}) \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F$ in **Step 5**.

Step 1. Let columns of $\widehat{V}_j \in \mathbb{O}^{m \times r_j}$ correspond to the leading r_j left singular vectors of the matrix $\widehat{\mathcal{B}}_j(l_j; l_1, \ldots, l_{j-1}, l_{j+1}, \ldots, l_d) \in \mathbb{R}^{m \times m^{d-1}}$, and $\widehat{V}_{j\perp} \in \mathbb{O}^{m \times (m-r_j)}$ correspond to the rest $m - r_j$ singular vectors of the same matrix. It follows that

$$\widehat{V}_{j\perp}\widehat{V}_{j\perp}^\top = I_m - \widehat{V}_j\widehat{V}_j^\top = I_m - \mathcal{P}_j^V.$$

Therefore

$$\begin{aligned}
&\|\mathcal{C}^* - \mathcal{C}^* \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F \\
=&\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top + \mathcal{C}^* \times_1 \mathcal{P}_1^V \times_2 \widehat{V}_{2\perp}\widehat{V}_{2\perp}^\top + \cdots + \mathcal{C}^* \times_1 \mathcal{P}_1^V \times \cdots \times_{d-1} \mathcal{P}_{d-1}^V \times_d \widehat{V}_{d\perp}\widehat{V}_{d\perp}^\top\|_F \\
\leq &\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F + \|\mathcal{C}^* \times_1 \mathcal{P}_1^V \times_2 \widehat{V}_{2\perp}\widehat{V}_{2\perp}^\top\|_F + \cdots + \|\mathcal{C}^* \times_1 \mathcal{P}_1^V \times \cdots \times_{d-1} \mathcal{P}_{d-1}^V \times_d \widehat{V}_{d\perp}\widehat{V}_{d\perp}^\top\|_F \\
\leq &\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F + \|\mathcal{C}^* \times_2 \widehat{V}_{2\perp}\widehat{V}_{2\perp}^\top\|_F \|\mathcal{P}_1^V\|_2 + \cdots + \|\mathcal{C}^* \times_d \widehat{V}_{d\perp}\widehat{V}_{d\perp}^\top\|_F \|\mathcal{P}_1^V\|_2 \cdots \|\mathcal{P}_{d-1}^V\|_2 \\
\leq &\sum_{j=1}^d \|\mathcal{C}^* \times_j \widehat{V}_{j\perp}\widehat{V}_{j\perp}^\top\|_F, \qquad \text{(A.27)}
\end{aligned}$$

where the last inequality follows from the fact that projection matrices have operator norm 1.

Step 2. We proceed to bound $\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F$, as $\|\mathcal{C}^* \times_1 \widehat{V}_{j\perp}\widehat{V}_{j\perp}^\top\|_F$ can be bounded

by the exact same calculations.

For $j \in \{1, \ldots, d\}$, let $U_j^* \in \mathbb{R}^{m \times r_j}$ denote the matrix with columns being the leading r_j singular vectors of $\mathcal{C}^*(l_j; l_1, \ldots, l_{j-1}, l_{j+1}, \ldots, l_d) \in \mathbb{R}^{m \times m^{d-1}}$. Let $\widehat{U}_j \in \mathbb{R}^{m \times r_j}$ be the the matrix with columns being the leading r_j singular vectors of $\mathcal{C}(\widehat{A} \times_j \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j})$. Here $\mathcal{C}(\widehat{A} \times_j \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j})$ is defined in (A.22). Then

$$\begin{aligned}
&\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F \\
\leq &\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 U_2^* U_2^{*\top}\|_F \qquad \text{(A.28)} \\
+ &\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 (I_m - U_2^* U_2^{*\top})\|_F. \qquad \text{(A.29)}
\end{aligned}$$

Note that

$$\text{(A.29)} \leq \|\mathcal{C}^* \times_2 (I_m - U_2^* U_2^{*\top})\|_F \|\widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_2 \leq \xi^*_{(r_1,\ldots,r_d)} + Cm^{-\alpha}, \qquad \text{(A.30)}$$

where the last inequality follows from Lemma 16 and the fact that $\|\widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_2 = 1$.
In addition, by applying Lemma 19 to the matrix $\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 U_2^* U_2^{*\top}(l_2; l_1, l_3, \ldots, l_d)$, it follows that

$$\text{(A.28)} \leq \sigma_{\min}^{-1}(\widehat{U}_2^\top U_2^*)\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 U_2^* U_2^{*\top}\widehat{U}_2\widehat{U}_2^\top\|_F$$

By Lemma 9, we have that $\sigma_{\min}^{-1}(\widehat{U}_2^\top U_2^*) = \mathrm{O}_{\mathbb{P}}(1)$. Consequently, there exists a constant C_1 such that with high probability,

$$\begin{aligned}
\text{(A.28)} \leq &C_1\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 U_2^* U_2^{*\top}\widehat{U}_2\widehat{U}_2^\top\|_F \\
\leq &C_1\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top\|_F + C_1\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 (I_m - U_2^* U_2^{*\top})\widehat{U}_2\widehat{U}_2^\top\|_F \\
\leq &C_1\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top\|_F + C_1\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 (I_m - U_2^* U_2^{*\top})\|_F\|\widehat{U}_2\widehat{U}_2^\top\|_2 \\
\leq &C_1\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top\|_F + C_1'(\xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}), \qquad \text{(A.31)}
\end{aligned}$$

where the last inequality follows from (A.30) and $\|\widehat{U}_2\widehat{U}_2^\top\|_2 = 1$. As a result

$$\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F \le C_1\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top\|_F + C_1'(\xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}). \tag{A.32}$$

Step 3. We bound $\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top\|_F$ in this step. Observe that

$$\begin{aligned}
&\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top\|_F \\
\le&\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times_3 U_3^*U_3^{*\top}\|_F &\text{(A.33)}\\
+&\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times_3 (I_m - U_3^*U_3^{*\top})\|_F. &\text{(A.34)}
\end{aligned}$$

Note that

$$\text{(A.34)} \le \|\mathcal{C}^*\times_3(I_m - U_3^*U_3^{*\top})\|_F\|\widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_2\|\widehat{U}_2\widehat{U}_2^\top\|_2 \le \xi^*_{(r_1,\ldots,r_d)} + Cm^{-\alpha}, \tag{A.35}$$

where the last inequality follows from Lemma 16.
In addition, by applying Lemma 19 to $\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times_3 U_3^*U_3^{*\top}(l_3; l_1, l_2, \ldots, l_d)$, it follows that

$$\text{(A.33)} \le \sigma_{\min}^{-1}(\widehat{U}_3^\top U_3^*)\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp} \times_2 \widehat{U}_2\widehat{U}_2^\top \times_3 U_3^*U_3^{*\top}\widehat{U}_3\widehat{U}_3^\top\|_F$$

By Lemma 9, we have that $\sigma_{\min}^{-1}(\widehat{U}_3^\top U_3^*) = \mathrm{O}_{\mathbb{P}}(1)$. Consequently, similar to (A.31), there exists a constant C_2 such that with high probability,

$$\text{(A.33)} \le C_2\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times_3 \widehat{U}_3\widehat{U}_3^\top\|_F + C_2(\xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}). \tag{A.36}$$

Combining (A.35), (A.36), and (A.32), it follows that

$$\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F \le C_2'\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times_3 \widehat{U}_3\widehat{U}_3^\top\|_F + C_2'(\xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}). \tag{A.37}$$

Step 4. Following the same argument that leads to (A.37), there exists a constant C' such that

$$\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F \le C'\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top\|_F + C'(\xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}). \tag{A.38}$$

It suffices to use Lemma 18 to control $\|\mathcal{C}^*\times_1\widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2\widehat{U}_2\widehat{U}_2^\top \times\cdots\times_d\widehat{U}_d\widehat{U}_d^\top\|_F$. Note that $\widehat{V}_1$ corresponds to the leading r_1 left singular vectors of the matrix $\widehat{\mathcal{B}}_1(l_1; l_2, \ldots, l_d) \in \mathbb{R}^{m\times m^{d-1}}$, where $\widehat{\mathcal{B}}_1 = \widehat{\mathcal{C}} \times_2 \widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top \in \mathbb{R}^{m^d}$. So

$$\begin{aligned}
&\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top\|_F \\
=&\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top(l_1; l_2, \ldots, l_d)\|_F \\
\le&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_2 \widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top(l_1; l_2, \ldots, l_d)\|_F \qquad \text{(A.39)} \\
+&\|\mathcal{C}^*\times_2\widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top(l_1; l_2, \ldots, l_d) - [\mathcal{C}^*\times_2\widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top(l_1; l_2, \ldots, l_d)]_{r_1}\|_F, \qquad \text{(A.40)}
\end{aligned}$$

where the inequality follows from Lemma 18. By $\|\widehat{U}_j\widehat{U}_j^\top - U_j^*U_j^{*\top}\| = \mathrm{O}_{\mathbb{P}}(m^{-1})$ for all $j \in \{1, \ldots, d\}$. Therefore by Lemma 14,

$$\begin{aligned}
\text{(A.39)} =&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_2 \widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top\|_F = \\
\le&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_2 \widehat{U}_2 \times \cdots \times_d \widehat{U}_d\|_F\|\widehat{U}_2^\top\|_2 \cdots \|\widehat{U}_d^\top\|_2 \\
=&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_2 \widehat{U}_2 \times \cdots \times_d \widehat{U}_d\|_F = \mathrm{O}_{\mathbb{P}}\left(\sqrt{\frac{mr_2\cdots r_d}{N}}\right). \qquad \text{(A.41)}
\end{aligned}$$

In addition by Lemma 17,

$$(\text{A.40}) \leq \xi^*_{(r_1,\ldots,r_d)} + C_3 m^{-\alpha}. \tag{A.42}$$

Observe that (A.41) and (A.42) together imply that with high probability,

$$\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top \times_2 \widehat{U}_2\widehat{U}_2^\top \times \cdots \times_d \widehat{U}_d\widehat{U}_d^\top\|_F \leq C_4\bigg(\sqrt{\frac{mr_2\cdots r_d}{N}} + \xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}\bigg). \tag{A.43}$$

(A.38) and (A.43) together imply that with high probability,

$$\|\mathcal{C}^* \times_1 \widehat{V}_{1\perp}\widehat{V}_{1\perp}^\top\|_F \leq C_5\bigg(\sqrt{\frac{mr_2\cdots r_d}{N}} + \xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}\bigg).$$

It follows from (A.27) that

$$\|\mathcal{C}^* - \mathcal{C}^*\times_1\mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F \leq \sum_{j=1}^d \|\mathcal{C}^* \times_j \widehat{V}_{j\perp}\widehat{V}_{j\perp}^\top\|_F \leq C_6\bigg(\sqrt{\frac{mr_2\cdots r_d}{N}} + \xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}\bigg). \tag{A.44}$$

Step 5. Note that the projection matrices $\mathcal{P}_j^V = \widehat{V}_j\widehat{V}_j^\top$ are computed based on the empirical measure $\widehat{A}$. Therefore $\{\mathcal{P}_j^V\}_{j=1}$ are independent of $\widehat{A}'$ and $\widehat{\mathcal{C}}' = \mathcal{C}(\widehat{A}' \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})$. So by Remark 6

$$\mathbb{E}(\|(\mathcal{C}^* - \widehat{\mathcal{C}}') \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F^2) = \mathrm{O}\left(\frac{\prod_{j=1}^d r_j}{N}\right).$$

Step 6. By (A.26), (A.44) and **Step 5**, it follows that with high probability,

$$\begin{aligned}\|\widetilde{\mathcal{C}} - \mathcal{C}^*\|_F \leq & \|\mathcal{C}^* - \mathcal{C}^* \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F + \|(\mathcal{C}^* - \widehat{\mathcal{C}}') \times_1 \mathcal{P}_1^V \times \cdots \times_d \mathcal{P}_d^V\|_F \\ \leq & C_7\bigg(\sqrt{\frac{mr_1r_2\cdots r_d}{N}} + \xi^*_{(r_1,\ldots,r_d)} + m^{-\alpha}\bigg). \end{aligned} \tag{A.45}$$

□

Lemma 9. *Suppose all the conditions as in Theorem 3 hold. Let U_j^* and $\widehat{U}_j$ be defined as in Lemma 10. Then $\|\widehat{U}_j^\top U_{j\perp}^*\|_2 = \mathrm{O}_{\mathbb{P}}\big(m^{-1}\big)$ and $\sigma_{\min}^{-1}(U_j^{*\top}\widehat{U}_j) = \mathrm{O}_{\mathbb{P}}(1)$.*

Proof. By Corollary 3, we have that

$$\|\widehat{U}_j^\top U_{j\perp}^*\|_2 = \sqrt{1 - \sigma_{\min}(\widehat{U}_j^\top U_j^*)} = \mathrm{O}_{\mathbb{P}}\bigg(\frac{\sigma^*_{j,r_j+1}}{\sigma^*_{j,r_j}} + \sqrt{\frac{\big(m + l_j^{d-1}\big)\log(N)}{N\sigma^{*2}_{j,r_j}}}\bigg).$$

Since (2.23) holds for sufficiently large constant, with the choices of m and ℓ_j in Theorem 3, it holds that $\|\widehat{U}_j^\top U_{j\perp}^*\|_2 = \mathrm{O}_{\mathbb{P}}\big(m^{-1}\big)$. Since with high probability $\sqrt{1 - \sigma^2_{\min}(\widehat{U}_j^\top U_j^*)} < 1$, it follows that $\sigma_{\min}^{-1}(\widehat{U}_j^\top U_j^*) = \mathrm{O}_{\mathbb{P}}(1)$. □

Consistency of Sketching

Suppose $\{\phi_k\}_{k=1}^\infty$ be a collection of basis function $L_2(\mathcal{O})$ defined in Assumption 8. For $j \in \{1,\ldots,d\}$, let $\mathcal{M}_j$ and $\mathcal{S}_j$ be defined as in (A.17) and (A.18) respectively. Denote

$$x = z_j \quad \text{and} \quad y = (z_1,\ldots,z_{j-1},\ldots,z_{j+1},\ldots,z_d).$$

Lemma 10. *For every $j \in \{1,\ldots,d\}$, suppose $\sigma^*_{j,r_j} \geq C_1 \max\left\{l^{-\alpha}, \sigma^*_{j,r_j+1}, \sqrt{\frac{(m+l^{d-1})\log(N)}{N}}\right\}$ for some absolute constant C_1. Let $U_j^* \in \mathbb{O}^{m\times r_j}$ with columns being the r_j-leading left singular vectors of the matrix $\mathcal{C}^*(l_j; l_1,\ldots,l_{j-1},l_{j+1},\ldots,l_d) \in \mathbb{R}^{m\times m^{d-1}}$, where $\mathcal{C}^* = A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \ldots \times_d \mathcal{P}_{\mathcal{M}_d} \in \mathbb{R}^{m^d}$. Let $\widehat{U}_j \in \mathbb{O}^{m\times r_j}$ with columns being the r_j-leading left*

singular vectors of the matrix $\mathcal{C}(\widehat{A} \times_x \mathcal{P}_{\mathcal{M}_j} \times_y \mathcal{P}_{\mathcal{S}_j}) \in \mathbb{R}^{m \times l_j^{d-1}}$. *Then*

$$\|U_j^* U_j^{*\top} - \widehat{U}_j \widehat{U}_j^\top\|_2 = \mathrm{O}_{\mathbb{P}}\Big(\frac{\sigma^*_{j,r_j+1}}{\sigma^*_{j,r_j}} + \sqrt{\frac{\big(m + l_j^{d-1}\big)\log(N)}{N\sigma^{*2}_{j,r_j}}} \Big).$$

Proof. Without loss of generality, assume $j = 1$. Throughout the proof, for brevity we denote $U^* = U_j^*$, $\widehat{U} = \widehat{U}_j$, $\mathcal{M} = \mathcal{M}_1$, $\mathcal{S} = \mathcal{S}_1$, $r = r_1$, $\ell = \ell_1$ and

$$\mathcal{M}^{\otimes d-1} = \mathcal{M}_2 \otimes \cdots \otimes \mathcal{M}_d.$$

It follows from (A.7) that $\mathcal{C}^*(l_1; l_2, \ldots, l_d)$ is the coefficient matrix of $\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})$. Therefore the columns of U^* are the r-leading left singular vectors of the matrix

$\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})$.

Step 1. Let $[A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r$ be the best rank-r approximation of $A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}$. See Theorem 14 for the definition of best rank r approximation for functions. In this step, we show that

$$\sigma_r([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}}) \geq \frac{1}{2}\sigma^*_{1,r}. \tag{A.46}$$

To see this, observe that

$$\begin{aligned}
&\| [A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \\
\leq &\| [A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \times_y \mathcal{P}_{\mathcal{S}} \|_2 \\
+ &\| A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \times_y \mathcal{P}_{\mathcal{S}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2.
\end{aligned}$$

Note that

$$
\begin{aligned}
&\| \, [A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}} - \; A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \times_y \mathcal{P}_{\mathcal{S}} \|_2 \\
=&\| \big([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \big) \times_y \mathcal{P}_{\mathcal{S}} \|_2 \\
\leq&\| [A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \|\mathcal{P}_{\mathcal{S}}\|_2 \\
\leq&\| [A]_r^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \|\mathcal{P}_{\mathcal{S}}\|_2 \\
=&\| \big([A]_r^* - A^* \big) \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \|\mathcal{P}_{\mathcal{S}}\|_2 \\
\leq&\| [A]_r^* - A^* \|_2 \|\mathcal{P}_{\mathcal{M}}\|_2 \|\mathcal{P}_{\mathcal{M}^{\otimes d-1}}\|_2 \|\mathcal{P}_{\mathcal{S}}\|_2 \leq \| [A]_r^* - A^* \|_2 = \sigma_{1,r+1}^*,
\end{aligned} \tag{A.47}
$$

where the second inequality follows from (A.10) and the fact that $[A]_r^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}$ is a function with rank at most r; the last inequality follows from the fact that projection functions has operator norm bounded by 1; and the last equality follows from Theorem 14.

In addition

$$
\begin{aligned}
&\| A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \times_y \mathcal{P}_{\mathcal{S}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \\
=&\| A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{S}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \\
\leq&\| A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{S}} - A^* \|_2 + \| A^* - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \\
=&\, \mathrm{O}(m^{-\alpha} + l^{-\alpha}) + \mathrm{O}(m^{-\alpha}) = \mathrm{O}(l^{-\alpha}),
\end{aligned}
$$

where the first equality follows from $\mathcal{S} \subset \mathcal{M}^{\otimes d-1}$; the second equality follows from Theorem 15; and the last equality holds because $l \leq m$. Therefore

$$
\| \, [A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}} \|_2 \leq \sigma_{1,r+1}^*(A^*) + C_2 l^{-\alpha} \tag{A.48}
$$

for some sufficiently large constant C_2. It follows that

$$\begin{aligned}
&\sigma_r([A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r)\\
\geq&\sigma_r(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}) - \| [A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}} - A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}\|_2\\
\geq&\sigma^*_{1,r} - C_3m^{-\alpha} - \sigma^*_{1,r+1}(A^*) - C_2l^{-\alpha} \geq \frac{1}{2}\sigma^*_{1,r},
\end{aligned}$$

where the first inequality follows from Lemma 23; the second inequality follows from Lemma 7 and (A.48); and the last inequality follows from the assumption that $\sigma^*_{1,r_1} \geq C_1 \max\{l^{-\alpha}, \sigma^*_{1,r_1+1}\}$.

Step 2. We show that the the column space of $\mathcal{C}([A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})$ equals to the column space of $[\mathcal{C}(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r$.

To see this, note that

$$\begin{aligned}
\mathcal{C}([A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}}) =&\mathcal{C}([A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r)\mathcal{C}(\mathcal{P}_{\mathcal{S}})\\
=&[\mathcal{C}(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r\mathcal{C}(\mathcal{P}_{\mathcal{S}}),
\end{aligned}$$

where the first equality follows from Lemma 5; and the second equality follows from Corollary 2 **(b)**. So the column space of $\mathcal{C}([A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})$ is contained in the column space of $[\mathcal{C}(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r$, or simply

$$\mathrm{Col}\big\{\mathcal{C}([A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})\big\} \subset \mathrm{Col}\big\{[\mathcal{C}(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r\big\}. \quad \text{(A.49)}$$

Note that the matrix $[\mathcal{C}(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r$ is at most rank r. By Lemma 7,

$$\sigma_r([\mathcal{C}(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r) = \sigma_r(\mathcal{C}(A^*\times_x\mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})) > 0.$$

So the matrix $[\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r$ is exactly rank r.

In addition, $\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})$ is at most rank r, because $[A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r$ is a function of rank at most r. By **Step 1**, $\sigma_r([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}}) > 0$.
So $\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})$ is exactly rank r.

Since both $[\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r$ and $\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})$ are exactly rank r matrices, (A.49) implies that

$$\mathrm{Col}\big\{\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})\big\} = \mathrm{Col}\big\{[\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r\big\}.$$

Step 3. The columns of U^* are the leading r left singular vectors of $\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})$.
Therefore the columns of U^* spans the column space of $[\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})]_r$.
By **Step 2**, the columns of U^* also spans the column space of $\mathrm{Col}\big\{\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})\big\}$. Note that columns of U^* are not necessarily the singular vectors of $\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})$, but $U^* {U^*}^\top$ is the projection matrix onto the column space of $\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})$.

It follows from Corollary 4 that

$$\|U^* {U^*}^\top - \widehat{U}\widehat{U}^\top\|_2 \le \sqrt{2}\frac{\text{(A.50)}}{\text{(A.51)} - \text{(A.50)}}, \quad \text{where}$$
$$\text{(A.50)} = \|\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}}) - \mathcal{C}(\widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{S}})\|_2$$
$$\text{(A.51)} = \sigma_r(\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})).$$

Note that with high probability,

$$\begin{aligned}(\text{A.50}) =& \|[A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}} - \widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{S}}\|_2 \\ \leq& \|[A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}} - A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{S}}\|_2 \\ &+ \|A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{S}} - \widehat{A} \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{S}}\|_2 \\ \leq& \sigma^*_{1,r+1} + C_2\bigg(\sqrt{\frac{\big(m + l^{d-1}\big)\log(N)}{N}}\bigg),\end{aligned}$$

where the last inequality follows from (A.47) and Lemma 11.

In addition,

$$(\text{A.51}) = \sigma_r(\mathcal{C}([A^* \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}]_r \times_y \mathcal{P}_{\mathcal{S}})) \geq \frac{1}{2}\sigma^*_{1,r}. \tag{A.52}$$

So

$$(\text{A.51}) - (\text{A.50}) \geq \frac{1}{2}\sigma^*_{1,r} - \sigma^*_{1,r+1} - C_2\bigg(\sqrt{\frac{(m + l^{d-1})\log(N)}{N}}\bigg) \geq \frac{1}{4}\sigma^*_{1,r},$$

where the last inequality holds because $\sigma^*_{1,r} \geq C_1 \max\left\{l^{-\alpha}, \sigma^*_{1,r+1}, \sqrt{\frac{(m+l^{d-1})\log(N)}{N}}\right\}$.

$$\|U^*U^{*\top} - \widehat{U}\widehat{U}^\top\|_2 \leq \sqrt{2}\frac{(\text{A.50})}{(\text{A.51}) - (\text{A.50})} \leq C_3\frac{\sigma^*_{1,r+1} + \sqrt{\frac{(m+l^{d-1})\log(N)}{N}}}{\sigma^*_{1,r}}.$$

□

Corollary 3. *Suppose all the conditions in Lemma 10 hold. Let $U^*_{j\perp} \in \mathbb{O}^{m\times(m-r_j)}$ be the orthogonal complement of U^*_j. Then*

$$\|\widehat{U}_j^\top U^*_{j\perp}\|_2 = \sqrt{1 - \sigma_{\min}(\widehat{U}_j^\top U^*_j)} = \mathrm{O}_{\mathbb{P}}\bigg(\frac{\sigma^*_{j,r_j+1}}{\sigma^*_{j,r_j}} + \sqrt{\frac{\big(m + l_j^{d-1}\big)\log(N)}{N\sigma^{*2}_{j,r_j}}}\bigg). \tag{A.53}$$

Here $\sigma_{\min}(\widehat{U}_j^\top U_j^*)$ *is the minimal singular value of the matrix* $\widehat{U}_j^\top U_j^* \in \mathbb{R}^{r\times r}$.

Proof. By Lemma 1 of [24],

$$\|\widehat{U}_j^\top U_{j\perp}^*\|_2 = \sqrt{1-\sigma_{\min}(\widehat{U}_j^\top U_j^*)} \le 2\|U_j^* U_j^{*\top} - \widehat{U}_j\widehat{U}_j^\top\|_2.$$

Therefore (A.53) is a direct consequence of the Lemma 10. □

A.4 Perturbation Bounds

Throughout this section, let $\{Z_i\}_{i=1}^N \subset \mathcal{O}^d$ be the i.i.d. data sampled from the density A^* and $\widehat{A} = \frac{1}{N}\sum_{i=1}^N \delta_{Z_i}$ be the corresponding empirical measure, and

$$\mathcal{C}^* = \mathcal{C}(A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d}) \quad \text{and} \quad \widehat{\mathcal{C}} = \mathcal{C}(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})$$

Lemma 11. *Let* $d_1 \in \mathbb{Z}^+$ *be such that* $d_1 < d$. *Suppose* $\{\phi_k\}_{k=1}^\infty$ *is a collection of* $L_2(\mathcal{O})$ *basis such that* $\|\phi_k\|_\infty \le C_\phi$ *for some absolute constant* C_ϕ. *For positive integers* m *and* ℓ, *denote*

$$\mathcal{M} = \mathrm{Span}\left\{\phi_{l_1}(z_1)\cdots\phi_{l_{d_1}}(z_{d_1})\right\}_{l_1,\ldots,l_{d_1}=1}^{m} \text{ and } \mathcal{N} = \mathrm{Span}\left\{\phi_{\eta_1}(z_{d_1+1})\cdots\phi_{\eta_{d-d_1}}(z_d)\right\}_{\eta_1,\ldots,\eta_{d-d_1}=1}^{\ell}. \tag{A.54}$$

Suppose in addition that $N \ge C_1 \max\{m^{d_1}, l^{d-d_1}\}$ *for some absolute constant* C_1. *Then it holds that*

$$\|(\widehat{A}-A^*)\times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{N}}\|_2 = \mathrm{O}_{\mathbb{P}}\left(\sqrt{\frac{\{m^{d_1}+l^{d-d_1}\}\log(N)}{N}}\right).$$

Proof. Denote

$$x = (z_1,\ldots,z_{d_1}) \quad \text{and} \quad y = (z_{d_1+1},\ldots,z_d).$$

For positive integers m and ℓ, by ordering the indexes $(l_1, \ldots, l_{d_1})$ and $(\eta_1, \ldots, \eta_{d-d_1})$ in (A.54), we can also write

$$\mathcal{M} = \text{span}\{\Phi_l(x)\}_{l=1}^{m^{d_1}} \quad \text{and} \quad \mathcal{N} = \text{span}\{\Psi_\eta(y)\}_{\eta=1}^{\ell^{d-d_1}}. \tag{A.55}$$

Here with the multi-index $\mu = (\mu_1, \ldots, \mu_p)$,

$$\Phi_\mu(x) = \phi_{l_1}(z_1) \ldots \phi_{l_p}(z_p).$$

So $\|\Phi_l\|_\infty \leq C_\phi^{d_1}$. Similarly $\|\Psi_\eta\|_\infty \leq C_\phi^{d-d_1}$. Therefore

$$\langle A^*, \Phi_l \otimes \Psi_\eta \rangle \leq C_\phi^d \|A^*\|_\infty \tag{A.56}$$

Step 1. Observe that $\widehat{A} \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}$ and $A^* \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}$ are both zero outside the subspace $\mathcal{M} \otimes \mathcal{N}$. Let $\mathcal{C}(\widehat{A} \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N})$ and $\mathcal{C}(A^* \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N})$ be the coefficient matrices in $\mathbb{R}^{m^{d_1} \times \ell^{d-d_1}}$. Note that

$$\begin{aligned} \|(\widehat{A} - A^*) \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}\|_2 &= \|\mathcal{C}(\widehat{A} \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}) - \mathcal{C}(A^* \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N})\|_2 \\ &= \|\frac{1}{N} \sum_{i=1}^N B_i - \mathbb{E}(B_i)\|_2, \end{aligned} \tag{A.57}$$

where the first equality follows from Lemma 4; and in the second equality, $B_i \in \mathbb{R}^{m^{d_1} \times l^{d-d_1}}$ is such that $(B_i)_{l,\eta} = \Phi_l \otimes \Psi_\eta(Z_i)$. Note that $\big(\mathcal{C}(\widehat{A} \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N})\big)_{l,\eta} = \frac{1}{N} \sum_{i=1}^N \Phi_\mu \otimes \Phi_\eta(Z_i)$. So

$$\frac{1}{N} \sum_{i=1}^N B_i = \mathcal{C}(\widehat{A} \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}).$$

Therefore $\mathbb{E}((B_i)_{l,\eta}) = \langle A^*, \Phi_l \otimes \Psi_\eta \rangle = \big(\mathcal{C}(A^* \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N})\big)_{l,\eta}$.

Step 2. In this step, we apply Theorem 16 to bound (A.57). To bound $\|B_i - \mathbb{E}(B_i)\|_2$, let $v = (v_1, \ldots, v_{m^{d_1}}) \in \mathbb{R}^{m^{d_1}}$ and $w = (w_1, \ldots, w_{\ell^{d-d_1}}) \in \mathbb{R}^{\ell^{d-d_1}}$. Then $\|B_i\|_2 = \sup_{|v|=1,|w|=1} v^\top B_i w$. Observe that

$$\begin{aligned} v^\top B_i w =& \sum_{l=1}^{m_1^d} \sum_{\eta=1}^{l^{d-d_1}} v_l (B_i)_{l,\eta} w_\eta = \sum_{l=1}^{m_1^d} \sum_{\eta=1}^{l^{d-d_1}} v_l \Phi_l \otimes \Psi_\eta(Z_i) w_\eta \\ \leq& \sum_{l=1}^{m_1^d} \sum_{\eta=1}^{l^{d-d_1}} |v_l| C_\phi^d |w_\eta| \leq C_\phi^d \sqrt{\sum_{l=1}^{m_1^d} |v_l|^2} \sqrt{\sum_{\eta=1}^{l^{d-d_1}} |w_\eta|^2} = C_\phi^d \sqrt{m_1^d l^{d-d_1}}, \end{aligned}$$

where the first inequality follows from the fact that $\|\Phi_l\|_\infty \leq C_\phi^{d_1}$ and $\|\Psi_\eta\|_\infty \leq C_\phi^{d-d_1}$. Similarly calculations shows that by (A.56), $\|\mathbb{E}(B_i)\|_2 \leq C_\phi^d \|A^*\|_\infty \sqrt{m_1^d l^{d-d_1}}$. So

$$\|B_i - \mathbb{E}(B_i)\|_2 \leq C_2 \sqrt{m_1^d l^{d-d_1}}.$$

Step 3. Note that $\mathbb{E}(\{B_i - \mathbb{E}(B_i)\}\{B_i - \mathbb{E}(B_i)\}^\top) = \mathbb{E}(B_i B_i^\top) - \mathbb{E}(B_i)\{\mathbb{E}(B_i)\}^\top$, and $\mathbb{E}(B_i)\{\mathbb{E}(B_i)\}^\top$ is positive semi-definite. So by 8.3.2 on page 125 of [213],

$$\|\mathbb{E}(\{B_i - \mathbb{E}(B_i)\}\{B_i - \mathbb{E}(B_i)\}^\top)\|_2 \leq \|\mathbb{E}(B_i B_i^\top)\|_2.$$

Let $v = (v_1, \ldots, v_{m^{d_1}}) \in \mathbb{R}^{m^{d_1}}$ be such that $|v| = 1$. Then $\|\mathbb{E}(B_i B_i^\top)\|_2 = \sup_{|v|=1} v^\top \mathbb{E}(B_i B_i^\top) v$.

Observe that

$$\begin{aligned}
v^\top \mathbb{E}(B_i B_i^\top) v &= \sum_{l,l'=1}^{m_1^d} \sum_{\eta=1}^{l^{d-d_1}} v_l \mathbb{E}\big\{\Phi_l \otimes \Psi_\eta(Z_i) \cdot \Phi_{l'} \otimes \Psi_\eta(Z_i)\big\} v_{l'} \\
&= \sum_{l,l'=1}^{m_1^d} \sum_{\eta=1}^{l^{d-d_1}} \int_{\mathcal{O}_1^d} \int_{\mathcal{O}^{d-d_1}} v_l \Phi_l(x) \Psi_\eta(y) \Phi_{l'}(x) \Psi_\eta(y) v_{l'} A^*(x,y) dx dy \\
&= \int_{\mathcal{O}^{d-d_1}} \int_{\mathcal{O}_1^d} \sum_{l=1}^{m_1^d} v_l \Phi_l(x) \sum_{l'=1}^{m_1^d} v_{l'} \Phi_{l'}(x) \sum_{\eta=1}^{l^{d-d_1}} \Psi_\eta^2(y) A^*(x,y) dx dy \\
&\leq \|A^*\|_\infty \int_{\mathcal{O}^{d-d_1}} \int_{\mathcal{O}_1^d} \Big\{ \sum_{l=1}^{m_1^d} v_l \Phi_l(x) \Big\}^2 \Big\{ \sum_{\eta=1}^{l^{d-d_1}} \Psi_\eta^2(y) \Big\} dx dy.
\end{aligned}$$

Since $\{\Psi_\eta(y)\}_{\eta=1}^{l^{d-d_1}}$ is a collection of orthonormal basis functions, we have

$$\int_{\mathcal{O}^{d-d_1}} \sum_{\eta=1}^{l^{d-d_1}} \Psi_\eta^2(y) dy = \sum_{\eta=1}^{l^{d-d_1}} \int_{\mathcal{O}^{d-d_1}} \Psi_\eta^2(y) dy = l^{d-d_1}.$$

Since $\{\Phi_\eta(y)\}_{l=1}^{m^{d-d_1}}$ is a collection of orthonormal basis functions, and $|v| = 1$,

$$\int_{\mathcal{O}_1^d} \Big\{ \sum_{l=1}^{m_1^d} v_l \Phi_l(x) \Big\}^2 dx = 1.$$

So

$$\|\mathbb{E}(\big\{B_i - \mathbb{E}(B_i)\big\}\big\{B_i - \mathbb{E}(B_i)\big\}^\top)\|_2 \leq \|\mathbb{E}(B_i B_i^\top)\|_2 \leq \|A^*\|_\infty l^{d-d_1}.$$

Step 4. Using similar calculations as in **Step 3**, it follows that

$$\|\mathbb{E}(\big\{B_i - \mathbb{E}(B_i)\big\}^\top \big\{B_i - \mathbb{E}(B_i)\big\})\|_2 \leq \|\mathbb{E}(B_i B_i^\top)\|_2 \leq \|A^*\|_\infty m^{d_1}.$$

Step 5. Therefore by Theorem 16, it holds that

$$\begin{aligned}
&\|(\widehat{A}-A^*)\times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{N}}\|_2 = \|\frac{1}{N}\sum_{i=1}^{N} B_i - \mathbb{E}(B_i)\|_2 \\
=&\mathrm{O}_{\mathbb{P}}\bigg(\sqrt{\frac{\{m^{d_1}+l^{d-d_1}\}\log(m^{d_1}+l^{d-d_1})}{N}} + \frac{\{m^{d_1}+l^{d-d_1}\}\log(m^{d_1}+l^{d-d_1})}{N}\bigg) \\
=&\mathrm{O}_{\mathbb{P}}\bigg(\sqrt{\frac{\{m^{d_1}+l^{d-d_1}\}\log(N)}{N}}\bigg),
\end{aligned}$$

where the last inequality follows from $N \geq C_1 \max\{m^{d_1}, l^{d-d_1}\}$, so that $m_1^d/N \leq \sqrt{m_1^d/N}$, and $l^{d-d_1}/N \leq \sqrt{l^{d-d_1}/N}$.

□

Theorem 16. *[Matrix Bernstein] Let $\{B_i\}_{i=1}^N$ be a set of independent random matrices with dimensions $m_1 \times m_2$. Suppose that for all i, $\mathbb{E}(B_i) = 0$ and $\|B_i\|_2 \leq L$. Denote $m = \max\{m_1, m_2\}$. Then for any $a > 2$, it holds that with probability greater than $1-2n^{-a+1}$ that*

$$\bigg\|\frac{1}{N}\sum_{i=1}^{N} B_i\bigg\|_2 \leq \sqrt{\frac{2a\nu\log(m)}{N}} + \frac{2aL\log(m)}{3N}.$$

Here $\nu = \max\big\{\|\mathbb{E}(B_i B_i^\top)\|_2, \|\mathbb{E}(B_i^\top B_i)\|_2\big\}$.

Proof. This is the well-known matrix Bernstein inequality. See [213] for the proof. □

Lemma 12. *Let $G \in L_2(\mathcal{O}^d)$ be any non-random functions. Then*

$$\mathbb{E}(\langle \widehat{A}-A^*, G\rangle^2) \leq \frac{\|G\|_{L_2(\mathcal{O}^d)}^2 \|A^*\|_\infty}{N}.$$

Proof. Note that $\langle \widehat{A}-A^*, G\rangle = \frac{1}{N}\sum_{i=1}^N \{G(Z_i) - \langle A^*, G\rangle\}$, and so $\mathbb{E}(\langle \widehat{A}-A^*, G\rangle) = 0$.

Consequently,

$$\mathbb{E}(\langle \widehat{A} - A^*, G\rangle^2) = \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N}\{G(Z_i) - \langle A^*, G\rangle\}\right) = \frac{1}{N}\mathrm{Var}\left(G(Z_1)\right)$$
$$\leq \frac{1}{N}\mathbb{E}(G^2(Z_1)) = \frac{1}{N}\int_{\mathcal{O}^d} G^2(z_1,\ldots,z_d)A^*(z_1,\ldots,z_d)dz_1\ldots dz_d \leq \frac{1}{N}\|G\|^2_{L_2(\mathcal{O}^d)}\|A^*\|_\infty.$$

□

Lemma 13. *Let $\mathcal{C}^* = \mathcal{C}(A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})$ and $\widehat{\mathcal{C}} = \mathcal{C}(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})$ be two tensors of size m^d. Let $\{Q_j\}_{j=1}^d \subset \mathbb{R}^{m\times q_j}$ be a collection of non-random matrix with $\|Q_j\|_2 \leq 1$. Then*

$$\mathbb{E}(\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_d Q_d\|_F^2) = \mathrm{O}\left(\frac{\prod_{j=1}^d q_j}{N}\right).$$

Remark 6. *Using the same notations as in Lemma 13, suppose in addition that $\{\widehat{Q}_j\}_{j=1}^d \subset \mathbb{R}^{m\times q_j}$ is a collection of random matrix with $\|\widehat{Q}_j\|_2 \leq 1$. Suppose in addition that $\{\widehat{Q}_j\}_{j=1}^d$ are independent of $\widehat{\mathcal{C}}$. Then*

$$\mathbb{E}(\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 \widehat{Q}_1 \times \cdots \times_d \widehat{Q}_d\|_F^2) = \mathrm{O}\left(\frac{\prod_{j=1}^d q_j}{N}\right).$$

The proof immediately follows from the argument of Lemma 13 and independence, and is therefore omitted for brevity.

Proof of Lemma 13. Let $Q_j = U_j\Sigma_j V_j^\top$ be the SVD of Q_j. Then

$$\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_d Q_d\|_F = \|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 U_1\Sigma_1 V_1^\top \times \cdots \times_d U_d\Sigma_d V_d^\top\|_F$$
$$= \|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 U_1\Sigma_1 \times \cdots \times_d U_d\Sigma_d\|_F \leq \|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 U_1 \times \cdots \times_d U_d\|_F\|\Sigma_1\|_2\cdots\|\Sigma_d\|_2$$
$$= \|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 U_1 \times \cdots \times_d U_d\|_F\|Q_1\|_2\cdots\|Q_d\|_2, \tag{A.58}$$

where the second equality follows from the fact that multiplication of orthogonal matrices are norm invariant; and the last inequality follows from $\|Q_j\|_2 = \|\Sigma_j\|_2$. For $j = 1, \ldots, d$, let $\beta_j \in \{1, \ldots, q_j\}$. Denote U_{j,l_j,β_j} the (l_j, β_j) entry of the matrix U_j. Then the $(\beta_1, \ldots, \beta_d)$ entry of $(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 U_1 \times \cdots \times_d U_d$ is

$$\begin{aligned}&\sum_{l_1,\ldots,l_d=1}^{m} U_{1,l_1,\beta_1} \cdots U_{d,l_d,\beta_d} \langle A^* - \widehat{A}, \phi_{l_1} \otimes \cdots \otimes \phi_{l_d} \rangle \\ =&\left\langle A^* - \widehat{A}, \sum_{l_1=1}^{m} U_{1,l_1,\beta_1} \phi_{l_1} \otimes \cdots \otimes \sum_{l_d=1}^{m} U_{d,l_d,\beta_d} \phi_{l_d} \right\rangle.\end{aligned}$$

Since the columns of U_j are orthogonal, it follows that $\left\| \sum_{l_j=1}^{m} U_{j,l_j,\beta_j} \phi_{l_j} \right\|_{L_2(\mathcal{O})} = 1$. So

$$\left\| \sum_{l_1=1}^{m} U_{1,l_1,\beta_1} \phi_{l_1} \otimes \cdots \otimes \sum_{l_d=1}^{m} U_{d,l_d,\beta_d} \phi_{l_d} \right\|_{L_2(\mathcal{O}^d)} = 1.$$

By Lemma 12,

$$\mathbb{E}\left(\left\langle A^* - \widehat{A}, \sum_{l_1=1}^{m} U_{1,l_1,\beta_1} \phi_{l_1} \otimes \cdots \otimes \sum_{l_d=1}^{m} U_{d,l_d,\beta_d} \phi_{l_d} \right\rangle^2 \right) \leq \frac{\|A^*\|_\infty}{N}.$$

Consequently,

$$\begin{aligned}&\mathbb{E}(\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 U_1 \times \cdots \times_d U_d\|_F^2) \\ =&\mathbb{E}\left(\sum_{\beta_1=1}^{q_1} \cdots \sum_{\beta_d=1}^{q_d} \left\langle A^* - \widehat{A}, \sum_{l_1=1}^{m} U_{1,l_1,\beta_1} \phi_{l_1} \otimes \cdots \otimes \sum_{l_d=1}^{m} U_{d,l_d,\beta_d} \phi_{l_d} \right\rangle^2 \right) = \frac{\|A^*\|_\infty q_1 \cdots q_d}{N}.\end{aligned}$$

The desired result follows from (A.58). □

Lemma 14. *Let $\mathcal{C}^* = \mathcal{C}(A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})$ and $\widehat{\mathcal{C}} = \mathcal{C}(\widehat{A} \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})$ be two tensors of size m^d. Let $\{Q_j\}_{j=1}^d \subset \mathbb{R}^{m \times q_j}$ be a collection of non-random matrices with $\|Q_j\|_2 \leq 1$. Let $\{W_j^*\}_{j=1}^d \subset \mathbb{O}^{m \times r_j}$ be a collection of nonrandom matrix and $\{W_{j\perp}^*\}_{j=1}^d \subset$*

$\mathbb{O}^{m\times(m-r_j)}$ *be the corresponding orthogonal complements. Let* $\{\widehat{W}_j\}_{j=1}^d \subset \mathbb{O}^{m\times r_j}$ *is a collection of random matrices such that*

$$\|(W_{j\perp}^*)^\top \widehat{W}_j\|_2^2 = \mathrm{O}_{\mathbb{P}}(m^{-1}). \tag{A.59}$$

Then for any $1 \le k \le d$*, it holds that*

$$\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{k-1} Q_{k-1} \times_k \widehat{W}_k \times \cdots \times_d \widehat{W}_d\|_F^2 = \mathrm{O}_{\mathbb{P}}\bigg(\frac{(\prod_{j=1}^{k-1} q_j)(\prod_{j=k}^{d} r_j)}{N}\bigg). \tag{A.60}$$

Proof. We proceed by induction. For $k = d$, (A.60) is shown in Lemma 13. Suppose (A.60) holds for $k = d_1 + 1$. It suffices to show (A.60) holds for $k = d_1$. Observe that

$$\begin{aligned}
&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{d_1-1} Q_{d_1-1} \times_{d_1} \widehat{W}_{d_1} \times \cdots \times_d \widehat{W}_d\|_F^2 \\
=&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{d_1-1} Q_{d_1-1} \times_{d_1} \{W_j^*(W_j^*)^\top + W_{j\perp}^*(W_{j\perp}^*)^\top\}\widehat{W}_{d_1} \times \cdots \times_d \widehat{W}_d\|_F^2 \\
\le&2\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{d_1-1} Q_{d_1-1} \times_{d_1} W_j^*(W_j^*)^\top \widehat{W}_{d_1} \times_{d_1+1} \widehat{W}_{d_1+1} \times \cdots \times_d \widehat{W}_d\|_F^2 \qquad \text{(A.61)} \\
&+2\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{d_1-1} Q_{d_1-1} \times_{d_1} W_{j\perp}^*(W_{j\perp}^*)^\top \widehat{W}_{d_1} \times_{d_1+1} \widehat{W}_{d_1+1} \times \cdots \times_d \widehat{W}_d\|_F^2, \qquad \text{(A.62)}
\end{aligned}$$

where the first equality follows from the fact that $W_j^*(W_j^*)^\top + W_{j\perp}^*(W_{j\perp}^*)^\top = I_m$, the identity matrix in $\mathbb{R}^{m\times m}$. Note that

$$\begin{aligned}
\text{(A.61)} \le&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{d_1-1} Q_{d_1-1} \times_{d_1} W_j^* \times_{d_1+1} \widehat{W}_{d_1+1} \times \cdots \times_d \widehat{W}_d\|_F^2 \|(W_j^*)^\top\|_2^2 \|\widehat{W}_{d_1}\|_2^2 \\
=&\|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{d_1-1} Q_{d_1-1} \times_{d_1} W_j^* \times_{d_1+1} \widehat{W}_{d_1+1} \times \cdots \times_d \widehat{W}_d\|_F^2 \\
=&\mathrm{O}_{\mathbb{P}}\bigg(\frac{(\prod_{j=1}^{d_1-1} q_j) r_p (\prod_{j=d_1+1}^{d} r_j)}{N}\bigg) = \mathrm{O}_{\mathbb{P}}\bigg(\frac{(\prod_{j=1}^{d_1-1} q_j)(\prod_{j=d_1}^{d} r_j)}{N}\bigg), \qquad \text{(A.63)}
\end{aligned}$$

where the second equality follows from the fact that (A.60) holds for $k = d_1$ and that projection matrices has operator norm bounded by 1. In addition,

$$\begin{aligned}(\text{A.62}) \leq & \|(\mathcal{C}^* - \widehat{\mathcal{C}}) \times_1 Q_1 \times \cdots \times_{d_1-1} Q_{d_1-1} \times_{d_1} W_{j\perp}^* \times_{d_1+1} \widehat{W}_{d_1+1} \times \cdots \times_d \widehat{W}_d\|_F^2 \|(W_{j\perp}^*)^\top \widehat{W}_{d_1}\|_2^2 \\ = & \mathrm{O}_{\mathbb{P}}\bigg(\frac{(\prod_{j=1}^{d_1-1} q_j)(m - r_p)(\prod_{j=d_1+1}^{d} r_j)}{N}\bigg)\mathrm{O}_{\mathbb{P}}(m^{-1}) = \mathrm{O}_{\mathbb{P}}\bigg(\frac{(\prod_{j=1}^{d_1-1} q_j)(\prod_{j=d_1}^{d} r_j)}{N}\bigg), \quad (\text{A.64})\end{aligned}$$

where the first equality follows from the fact that (A.60) holds for $k = d_1$ and (A.59); and the last equality follows from the observation that $\frac{m-r_p}{m} \leq 1 \leq r_p$. The case of $k = d_1 + 1$ follows from (A.63), (A.64). □

Lemma 15. *Let* $\{\sigma_{j,\rho}^*\}_{\rho=r_j+1}^{\infty}$ *be defined as in* (A.19)*, and* $\xi_{(r_1,\ldots,r_d)}^*$ *be defined in* (A.25)*. Then for all* $j \in \{1, \ldots, d\}$*, it holds that*

$$\sqrt{\sum_{\rho=r_j+1}^{\infty} \sigma_{j,\rho}^{*2}} \leq \xi_{(r_1,\ldots,r_d)}^*.$$

Proof. Denote $x = z_j$ and $y = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d)$. By the definition of $\xi_{(r_1,\ldots,r_d)}^*$, there exists a sequence $B_k \in L_2(\mathcal{O}^d)$ with $k \in \mathbb{Z}^+$ such that $\mathrm{Rank}_j(B_k) \leq r_j$ for $j = \{1, \ldots, d\}$, and that

$$\lim_{k\to\infty} \|A^*(x, y) - B_k(x, y)\|_{L_2(\mathcal{O}^d)} = \xi_{(r_1,\ldots,r_d)}^*.$$

See Definition 3 for the definition of Rank_j. Then since $\mathrm{Rank}_j(B_k) \leq r_j$, it follows that the two-variable function $B_k(x, y)$ has rank at most r_j. Therefore by (A.10),

$$\|A^*(x, y) - [A^*]_{r_j}(x, y)\|_{L_2(\mathcal{O}^d)} \leq \|A^*(x, y) - B_k(x, y)\|_{L_2(\mathcal{O}^d)}. \quad (\text{A.65})$$

Since $\|A^*(x,y) - [A^*]_{r_j}(x,y)\|^2_{L_2(\mathcal{O}^d)} = \sum_{\rho=r_j+1}^{\infty} \sigma^{*2}_{j,\rho}$, (A.65) implies that

$$\sqrt{\sum_{\rho=r_j+1}^{\infty} \sigma^{*2}_{j,\rho}} \leq \lim_{k\to\infty} \|A^*(x,y) - B_k(x,y)\|_{L_2(\mathcal{O}^d)} = \xi^*_{(r_1,\ldots,r_d)}.$$

□

Lemma 16. *Let $\mathcal{C}^* = \mathcal{C}(A^* \times_1 \mathcal{P}_{\mathcal{M}_1} \times \cdots \times_d \mathcal{P}_{\mathcal{M}_d})$. For $j \in \{1,\ldots,d\}$, let $U^*_j \in \mathbb{R}^{m\times r_j}$ denote the matrix with columns being the leading r_j singular vectors of $\mathcal{C}^*(l_j; l_1,\ldots,l_{j-1},l_{j+1},\ldots,l_d) \in \mathbb{R}^{m\times m^{d-1}}$. Then*

$$\|\mathcal{C}^* \times_j (I_m - U^*_j U^{*\top}_j)\|_F \leq \xi^*_{(r_1,\ldots,r_d)} + Cm^{-\alpha},$$

where $\xi^*_{(r_1,\ldots,r_d)}$ is defined in (A.25).

Proof. Without loss of generality, assume $j = 1$. Denote $x = z_1$ and $y = (z_2,\ldots,z_d)$, and $\mathcal{C}^*_1 = \mathcal{C}^*(l_1; l_2,\ldots,l_d)$. Then from the definition, it follows that $\mathcal{C}^*_1$ is the coefficient matrix of $\mathcal{C}(A^* \times_x \mathcal{P}_{\mathcal{M}_1} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})$, where

$$\mathcal{M}^{\otimes d-1} = \mathcal{M}_2 \otimes \cdots \otimes \mathcal{M}_d.$$

We have $\|\mathcal{C}^* \times_1 (I_m - U^*_1 U^{*\top}_1)\|_F = \|(I_m - U^*_1 U^{*\top}_1)\mathcal{C}^*_1\|_F = \|\mathcal{C}^*_1 - [\mathcal{C}^*_1]_{r_1}\|_F$. Note that

$$\|\mathcal{C}^*_1 - [\mathcal{C}^*_1]_{r_1}\|_F = \sum_{\rho=r_1+1}^{\infty} \sigma^2_\rho(\mathcal{C}^*_1) = \sum_{\rho=r_1+1}^{\infty} \sigma^2_\rho(A^* \times_x \mathcal{P}_{\mathcal{M}_1} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}) \tag{A.66}$$

where the second equality follows from Corollary 2.

We are in the position to apply Lemma 24, the Mirsky's inequality in Hilbert space. Observe

that

$$\begin{aligned}
\|\mathcal{C}_1^* - [\mathcal{C}_1^*]_{r_1}\|_F &= \sqrt{\sum_{\rho=r_1+1}^{\infty} \sigma_\rho^2(A^* \times_x \mathcal{P}_{\mathcal{M}_1} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})} \\
=&\sqrt{\sum_{\rho=r_1+1}^{\infty} \sigma_\rho^2(A^* \times_x \mathcal{P}_{\mathcal{M}_1} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}})} \pm \sqrt{\sum_{\rho=r_1+1}^{\infty} \sigma_\rho^2(A^*(x,y))} \\
\leq&\sqrt{\sum_{\rho=r_1+1}^{\infty} \left\{\sigma_\rho(A^* \times_x \mathcal{P}_{\mathcal{M}_1} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}) - \sigma_\rho(A^*(x,y))\right\}^2} + \sqrt{\sum_{\rho=r_1+1}^{\infty} \sigma_\rho^2(A^*(x,y))},
\end{aligned}$$

where the inequality follows from the triangle inequality. We have that

$$\begin{aligned}
&\sum_{\rho=r_1+1}^{\infty} \left\{\sigma_\rho(A^* \times_x \mathcal{P}_{\mathcal{M}_1} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}) - \sigma_\rho(A^*(x,y))\right\}^2 \\
\leq&\|A^* - A^* \times_x \mathcal{P}_{\mathcal{M}_1} \times_y \mathcal{P}_{\mathcal{M}^{\otimes d-1}}\|_{L_2(\mathcal{O}^d)}^2 \leq Cm^{-2\alpha},
\end{aligned}$$

where the first inequality follows from Lemma 24; and the second inequality follows from Theorem 15 with C being a sufficiently large constant. By viewing $A^*(x,y)$ as a function of (x,y), from (A.19) the singular values of $A^*(x,y)$ is $\{\sigma_{1,\rho}^*\}_{\rho=1}^{\infty}$. So $\sum_{\rho=r_1+1}^{\infty} \sigma_\rho^2(A^*(x,y)) = \sum_{\rho=r_1+1}^{\infty} \sigma_{1,\rho}^{*2}$. Therefore

$$\|\mathcal{C}_1^* - [\mathcal{C}_1^*]_{r_1}\|_F \leq \sqrt{C}m^{-\alpha} + \sqrt{\sum_{\rho=r_1+1}^{\infty} \sigma_{1,\rho}^{*2}} = \sqrt{C}m^{-\alpha} + \sqrt{\sum_{\rho=r_1+1}^{\infty} \sigma_{1,\rho}^{*2}} \leq \sqrt{C}m^{-\alpha} + \xi_{(r_1,\ldots,r_d)}^*, \tag{A.67}$$

where the last inequality follows from Lemma 15.

□

Lemma 17. *Let $\{Q_j\}_{j=2}^d \subset \mathbb{R}^{m\times m}$ be an arbitrary collection matrices such that $\|Q_j\|_2 \leq 1$.*

Then

$$\|\mathcal{C}^* \times_2 Q_2 \times \cdots \times_d Q_d(l_1; l_2, \ldots, l_d) - [\mathcal{C}^* \times_2 Q_2 \times \cdots \times_d Q_d(l_1; l_2, \ldots, l_d)]_{r_1}\|_F \quad \text{(A.68)}$$
$$\leq \xi^*_{(r_1,\ldots,r_d)} + Cm^{-\alpha},$$

where $\xi^*_{(r_1,\ldots,r_d)}$ *is defined in* (A.25).

Proof. Denote $\mathcal{C}_1^* = \mathcal{C}^*(l_1; l_2, \ldots, l_d)$. Then

$$\mathcal{C}^* \times_2 Q_2 \times \cdots \times_d Q_d(l_1; l_2, \ldots, l_d) = \mathcal{C}_1^*(Q_2 \otimes \cdots \otimes Q_d)(l_1; l_2, \ldots, l_d),$$

where $Q_2 \otimes \cdots \otimes Q_d$ is a matrix of size $m^{d-1} \times m^{d-1}$. Since $[\mathcal{C}_1^*]_{r_1}$ has rank at most r_1, $[\mathcal{C}_1^*]_{r_1}(Q_2 \otimes \cdots \otimes Q_d)$ has rank at most r_1. So

$$\begin{aligned}
\text{(A.68)} =& \|\mathcal{C}_1^*(Q_2 \otimes \cdots \otimes Q_d) - [\mathcal{C}_1^*(Q_2 \otimes \cdots \otimes Q_d)]_{r_1}\|_F \leq \|\mathcal{C}_1^*(Q_2 \otimes \cdots \otimes Q_d) - [\mathcal{C}_1^*]_{r_1}(Q_2 \otimes \cdots \otimes Q_d)\|_F \\
=& \|(\mathcal{C}_1^* - [\mathcal{C}_1^*]_{r_1})(Q_2 \otimes \cdots \otimes Q_d)\|_F \leq \|\mathcal{C}_1^* - [\mathcal{C}_1^*]_{r_1}\|_F \|Q_2 \otimes \cdots \otimes Q_d\|_2 \\
\leq& \|\mathcal{C}_1^* - [\mathcal{C}_1^*]_{r_1}\|_F \leq \xi^*_{(r_1,\ldots,r_d)} + Cm^{-\alpha}
\end{aligned}$$

where the first inequality follows from (A.10); the second inequality follows from $\|Q_2 \otimes \cdots \otimes Q_d\|_2 \leq \|Q_1\|_2 \cdots \|Q_d\|_2 \leq 1$; and the last inequality follows from (A.67). □

Perturbation bounds for matrices

Let $\mathcal{C} \in \mathbb{R}^{m \times l}$ be any matrix. Denote $k = \min\{m, l\}$. Then the singular value decomposition of $\mathcal{C}$ satisfies

$$\mathcal{C} = U \Sigma V^\top.$$

Here $U \in \mathbb{O}^{m \times k}$ with columns being the left singular vectors of $\mathcal{C}$; $V \in \mathbb{O}^{l \times k}$ with columns being the right singular vectors of $\mathcal{C}$; and $\Sigma \in \mathbb{R}^{k \times k}$ is a diagonal matrix with diagonal entries $\sigma_1(\mathcal{C}) \geq \sigma_2(\mathcal{C}) \geq \ldots \geq 0$ being the singular values of $\mathcal{C}$.

Let $r \leq k$ be a positive integer. Suppose that $U_r \in \mathbb{O}^{m\times r}, V_r \in \mathbb{O}^{l\times r}$, $U_{r\perp} \in \mathbb{O}^{m\times(k-r)}$, and $V_{r\perp} \in \mathbb{O}^{l\times(k-r)}$ are such that

$$[U_r, U_{r\perp}] = U \quad \text{and} \quad [V_r, V_{r\perp}] = V.$$

Then U_r has columns being the first r leading singular vectors of $\mathcal{C}$. Let $\Sigma_r \in \mathbb{R}^{r\times r}$ be the diagonal matrix whose diagonal entries are $\{\sigma_1(\mathcal{C}), \ldots, \sigma_r(\mathcal{C})\}$, and $\Sigma_{r\perp} \in \mathbb{R}^{k-r\times k-r}$ be the diagonal matrix whose diagonal entries are $\{\sigma_{r+1}(\mathcal{C}), \ldots, \sigma_k(\mathcal{C})\}$. Then

$$\mathcal{C} = U_r \Sigma_r V_r^\top + U_{r\perp} \Sigma_{r\perp} V_{r\perp}^\top. \tag{A.69}$$

Denote

$$[\mathcal{C}]_r = U_r \Sigma_r V_r^\top.$$

Then $[\mathcal{C}]_r$ is the best rank r approximation of $\mathcal{C}$ in the sense that

$$\|\mathcal{C} - [\mathcal{C}]_r\|_2 = \min_{B\in\mathbb{R}^{m\times l},\ \mathrm{rank}(B)\leq r} \|\mathcal{C} - B\|_2 = \sigma_{r+1}(\mathcal{C}),$$
$$\|\mathcal{C} - [\mathcal{C}]_r\|_F = \min_{B\in\mathbb{R}^{m\times l},\ \mathrm{rank}(B)\leq r} \|\mathcal{C} - B\|_F = \sigma_{r+1}(\mathcal{C}).$$

Theorem 17. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two matrices of dimensions $n_1 \times n_2$. Denote $n = \min\{n_1, n_2\}$. Let $\{\sigma_j(\mathcal{C})\}_{j=1}^n$ be the singular values of $\mathcal{C}$ in the decreasing order, and $\{\sigma_j(\mathcal{C}')\}_{j=1}^n$ be the singular values of $\mathcal{C}'$ in the decreasing order.*

(a) *(Weyl's inequality) For all $j \in \{1, \ldots, n\}$, it holds that $|\sigma_j(\mathcal{C}) - \sigma_j(\mathcal{C}')| \leq \|\mathcal{C} - \mathcal{C}'\|_2$.*

(b) *(Mirsky's inequality) It holds that*

$$\sum_{j=1}^n (\sigma_j(\mathcal{C}) - \sigma_j(\mathcal{C}'))^2 \leq \|\mathcal{C} - \mathcal{C}'\|_F^2 = \sum_{j=1}^n \sigma_j^2(\mathcal{C} - \mathcal{C}'),$$

where $\sigma_j(\mathcal{C} - \mathcal{C}')$ denotes the j-th singular value of $\mathcal{C} - \mathcal{C}'$.

Proof. See [97] for the proofs of both results. □

Lemma 18. *Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be three matrices of size $m \times l$ and that $\mathcal{C} = \mathcal{A} + \mathcal{B}$. Suppose for $r \leq \min\{m, l\}$, the SVD of $\mathcal{C}$ satisfies*

$$\mathcal{C} = U_r \Sigma_r V_r + U_{r\perp} \Sigma_{r\perp} V_{r\perp}$$

as in (A.69)*. Then*

$$\|U_{r\perp} U_{r\perp}^\top \mathcal{A}\|_F \leq 2\|\mathcal{B}\|_F + \|\mathcal{A} - [\mathcal{A}]_r\|_F.$$

Proof. Note that

$$\|U_{r\perp} U_{r\perp}^\top \mathcal{A}\|_F = \|U_{r\perp} U_{r\perp}^\top (\mathcal{C} - \mathcal{B})\|_F \leq \|U_{r\perp} U_{r\perp}^\top \mathcal{C}\|_F + \|U_{r\perp} U_{r\perp}^\top \mathcal{B}\|_F. \tag{A.70}$$

Observe that

$$\begin{aligned}\|U_{r\perp} U_{r\perp}^\top \mathcal{C}\|_F =& \|U_{r\perp} U_{r\perp}^\top \Sigma_{r\perp} V_{r\perp}\|_F = \|U_{r\perp}^\top \Sigma_{r\perp} V_{r\perp}\|_F = \|\mathcal{C} - [\mathcal{C}]_r\|_F \\ \leq& \|\mathcal{C} - [\mathcal{A}]_r\|_F \leq \|\mathcal{C} - \mathcal{A}\|_F + \|\mathcal{A} - [\mathcal{A}]_r\|_F = \|\mathcal{B}\|_F + \|\mathcal{A} - [\mathcal{A}]_r\|_F,\end{aligned}$$

where the first equality follows from the fact that $\mathcal{U}_{r\perp}^\top \mathcal{U}_r = 0$; and the second inequality follows from Lemma 20. In addition,

$$\|U_{r\perp} U_{r\perp}^\top \mathcal{B}\|_F \leq \|U_{r\perp}\|_2 \|U_{r\perp}^\top\|_2 \|\mathcal{B}\|_F = \|\mathcal{B}\|_F,$$

where the last equality follows from the fact that $\|U_{r\perp}\|_2 = 1$. The desired result directly follows from (A.70). □

Lemma 19. *Let $\mathcal{C} \in \mathbb{R}^{m\times n}$ and $V, U \in \mathbb{O}^{m\times r}$. Denote $\mathcal{P}_V = VV^\top$ and $\mathcal{P}_U = UU^\top$. Then*

$$\|\mathcal{P}_V \mathcal{C}\|_F \leq \sigma_{\min}^{-1}(U^\top V) \|\mathcal{P}_U \mathcal{P}_V \mathcal{C}\|_F.$$

Proof. Note that

$$\begin{aligned}\|\mathcal{P}_U\mathcal{P}_V\mathcal{C}\|_F =&\|UU^\top VV^\top\mathcal{C}\|_F = \|U^\top VV^\top\mathcal{C}\|_F \\ \geq&\sigma_{\min}(U^\top V)\|V^\top\mathcal{C}\|_F = \sigma_{\min}(U^\top V)\|VV^\top\mathcal{C}\|_F = \sigma_{\min}(U^\top V)\|\mathcal{P}_V\mathcal{C}\|_F,\end{aligned}$$

where the second equality follows from Lemma 20; the inequality follows from Lemma 21; and the third equality follows from Lemma 20. □

Lemma 20. *Suppose $U \in \mathbb{O}^{m\times r}$ and $M \in \mathbb{R}^{m\times n}$. Then $\|UU^\top M\|_F = \|U^\top M\|_F$.*

Proof. Observe that

$$\|UU^\top M\|_F^2 = \mathrm{Tr}(UU^\top MM^\top UU^\top) = \mathrm{Tr}(U^\top UU^\top MM^\top U) = \mathrm{Tr}(U^\top MM^\top U) = \|U^\top M\|_F^2$$

□

Lemma 21. *Let $\mathcal{A} \in \mathbb{R}^{m\times m}$ and $\mathcal{B} \in \mathbb{R}^{m\times n}$ be two matrices. Then*

$$\sigma_{\min}(\mathcal{A})\|\mathcal{B}\|_F \leq \|\mathcal{A}\mathcal{B}\|_F.$$

Proof. Denote the SVD of A as $\mathcal{A} = \sum_{j=1}^m \sigma_j(\mathcal{A})v_j w_j^\top$. Then

$$\|\mathcal{A}\mathcal{B}\|_F^2 = \|\sum_{j=1}^m v_j\{\sigma_j(A)w_j^\top\mathcal{B}\}\|_F^2 = \sum_{j=1}^m \|\sigma_j(\mathcal{A})w_j^\top\mathcal{B}\|_2^2 \geq \sigma_{min}^2(\mathcal{A})\sum_{j=1}^m \|w_j^\top\mathcal{B}\|_2^2 = \sigma_{min}^2(\mathcal{A})\|\mathcal{B}\|_F^2,$$

where the second equality follows from the fact that $\{v_j\}_{j=1}^m$ is a set of complete basis vectors; and the last equality follows from the fact that $\{w_j\}_{j=1}^m$ is a set of complete basis vectors. □

Theorem 18 (Wedin)**.** *Suppose without loss of generality that $n_1 \leq n_2$. Let $M = M^* +$*

E, M^* be two matrices in $\mathbb{R}^{n_1 \times n_2}$ whose svd are given respectively by

$$M^* = \sum_{i=1}^{n_1} \sigma_i^* u_i^* v_i^{*\top} \quad \textit{and} \quad M = \sum_{i=1}^{n_1} \sigma_i u_i v_i^{\top}$$

where $\sigma_1^* \geq \cdots \geq \sigma_{n_1}^*$ and $\sigma_1 \geq \cdots \geq \sigma_{n_1}$. For any $r \leq n_1$, let

$$\Sigma^* = diag([\sigma_1^*, \cdots, \sigma_r^*]) \in \mathbb{R}^{r \times r}, \quad U^* = [u_1^*, \cdots, u_r^*] \in \mathbb{O}^{n_1 \times r}, \quad V = [v_1^*, \cdots, v_r^*] \in \mathbb{O}^{n_2 \times r},$$
$$\Sigma = diag([\sigma_1, \cdots, \sigma_r]) \in \mathbb{R}^{r \times r}, \quad U = [u_1, \cdots, u_r] \in O^{n_1 \times r}, \quad V = [v_1, \cdots, v_r] \in O^{n_2 \times r}.$$

Denote $\mathcal{P}_{U^*} = U^* U^{*\top}$ and $\mathcal{P}_U = UU^{\top}$. If $\|E\|_2 < \sigma_r^* - \sigma_{r+1}^*$, then

$$\|\mathcal{P}_{U^*} - \mathcal{P}_U\|_2 \leq \frac{\sqrt{2}\|E\|_2}{\sigma_r^* - \sigma_{r+1}^* - \|E\|_2}. \tag{A.71}$$

Proof. This is the well-known Wedin's theorem. See section 2 of [39] for proofs. □

Corollary 4 (Wedin)**.** *Suppose all the conditions in Theorem 18 hold. Suppose in addition that both M^* and M are exactly rank r. Let $W^* \in \mathbb{O}^{n_1 \times r}$ be such that the columns of W^* spans the column space of M^*, and $W \in \mathbb{O}^{n_1 \times r}$ be such that the columns of W spans the column space of M, Then*

$$\|W^* W^{*\top} - WW^{\top}\|_2 \leq \frac{\sqrt{2}\|E\|_2}{\sigma_r^* - \|E\|_2}.$$

Proof. It suffices to observe that $W^* W^{*\top} = U^* U^{*\top}$ and $WW^{\top} = UU^{\top}$, and $\sigma_{r+1}^* = 0$. □

Corollary 5. *Let $\mathcal{W}$ and $\mathcal{W}'$ be two Hilbert spaces. Let M and E be two finite rank operators on $\mathcal{W} \otimes \mathcal{W}'$ and denote $M = M^* + E$. Let the SVD of M^* and M are given respectively by*

$$M^* = \sum_{i=1}^{r_1} \sigma_i^* u_i^* \otimes v_i^* \quad \textit{and} \quad M = \sum_{i=1}^{r_2} \sigma_i u_i \otimes v_i$$

where $\sigma_1^* \geq \cdots \geq \sigma_{r_1}^*$ *and* $\sigma_1 \geq \cdots \geq \sigma_{r_2}$. *For* $r \leq min\{r_1, r_2\}$, *denote*

$$U^* = \mathrm{Span}(\{u_i^*\}_{i=1}^r) \quad \textit{and} \quad U = \mathrm{Span}(\{u_i\}_{i=1}^r)$$

Let $\mathcal{P}_{U^*}$ *to be projection operator from* $\mathcal{W}$ *to* U^*, *and* $\mathcal{P}_U$ *to be projection operator from* $\mathcal{W}$ *to* U. *If* $\|E\|_2 < \sigma_r^* - \sigma_{r+1}^*$, *then*

$$\|\mathcal{P}_{U^*} - \mathcal{P}_U\|_2 \leq \frac{\sqrt{2}\|E\|_2}{\sigma_r^* - \sigma_{r+1}^* - \|E\|_2}.$$

Proof. Let

$$\mathcal{S} = \mathrm{Span}(\{u_i^*\}_{i=1}^{r_1}, \{u_i\}_{i=1}^{r_2}) \quad \text{and} \quad \mathcal{S}' = \mathrm{Span}(\{v_i^*\}_{r=1}^{r_1}, \{v_i\}_{i=1}^{r_2}).$$

Then M^* and M can be viewed as finite-dimensional matrices on $\mathcal{S} \otimes \mathcal{S}'$. Since $\mathcal{P}_{U^*} = \sum_{i=1}^r u_i^* \otimes u_i^*$ and $\mathcal{P}_U = \sum_{i=1}^r u_i \otimes u_i$, the desired result follows from Theorem 18.

□

Perturbation Bounds in Function Spaces

Lemma 22. *Let* A *and* B *be two compact self-adjoint operators on a Hilbert space* $\mathcal{W}$. *Denote* $\lambda_k(A)$ *and* $\lambda_k(A+B)$ *to be the* k*-th eigenvalue of* A *and* $A+B$ *respectively. Then*

$$|\lambda_k(A+B) - \lambda_k(A)| \leq \|B\|_2.$$

Proof. By the min-max principle, for any compact self-adjoint operators H and any $S_k \subset \mathcal{W}$ being a k-dimensional subspace

$$\max_{S_k} \min_{x \in S_k, \|x\|_{\mathcal{W}}=1} H[x,x] = \lambda_k(H).$$

It follows that

$$\begin{aligned}\lambda_k(A+B) =& \max_{S_k} \min_{x\in S_k, \|x\|_{\mathcal{W}}=1} (A+B)[x,x] \\ \leq& \max_{S_k} \min_{x\in S_k, \|x\|_{\mathcal{W}}=1} A[x,x] + \|B\|_2 \|x\|_{\mathcal{W}}^2 \\ =& \lambda_k(A) + \|B\|_2.\end{aligned}$$

The other direction follows from symmetry. □

Lemma 23. *Let $\mathcal{W}$ and $\mathcal{W}'$ be two separable Hilbert spaces. Suppose A and B are two compact operators from $\mathcal{W} \otimes \mathcal{W}' \to \mathbb{R}$. Then*

$$\left|\sigma_k(A+B) - \sigma_k(A)\right| \leq \|B\|_2.$$

Proof. Let $\{\phi_i\}_{i=1}^{\infty}$ and $\{\phi_i'\}_{i=1}^{\infty}$ be the orthogonal basis of $\mathcal{W}$ and $\mathcal{W}'$. Let

$$\mathcal{W}_j = \mathrm{Span}(\{\phi_i\}_{i=1}^{j}) \quad \text{and} \quad \mathcal{W}_j' = \mathrm{Span}(\{\phi_i'\}_{i=1}^{j}).$$

Denote

$$A_j = A \times \mathcal{P}_{\mathcal{W}_j} \times \mathcal{P}_{\mathcal{W}_j'} \quad \text{and} \quad (A+B)_j = (A+B) \times \mathcal{P}_{\mathcal{W}_j} \times \mathcal{P}_{\mathcal{W}_j'}.$$

Note that $(A+B)_j = A_j + B_j$ due to linearity. Since A and $A+B$ are compact,

$$\lim_{j\to\infty} \|A - A_j\|_F = 0 \quad \text{and} \quad \lim_{j\to\infty} \|(A+B) - (A+B)_j\|_F = 0.$$

Then $AA^\top$ and $A_jA_j^\top$ are two compact self-adjoint operators on $\mathcal{W}$ and that

$$\lim_{j\to\infty} \|AA^\top - A_jA_j^\top\|_F = 0.$$

By Lemma 22, $\lim_{j\to\infty} \lambda_k(A_jA_j^\top) = \lambda_k(AA^\top)$. Observe that Since $\sigma_k(A_j)$ and $\sigma_k(A)$ are

both positive, that $\lambda_k(AA^\top) = \sigma_k^2(A)$, and that $\lambda_k(A_j A_j^\top) = \sigma_k^2(A_j)$. It follows that

$$\lim_{j\to\infty} \sigma_k(A_j) = \sigma_k(A).$$

Similarly $\lim_{j\to\infty} \sigma_k((A+B)_j) = \sigma_k(A+B)$. By the finite dimensional SVD perturbation theory (see Theorem 3.3.16 on page 178 of [96]), it follows that

$$\left|\sigma_k((A+B)_j) - \sigma_k(A_j)\right| \le \|B_j\|_2 \le \|B\|_2.$$

The desired result follows by taking the limit as $j \to \infty$.

□

Lemma 24 (Mirsky in Hilbert space)**.** *Suppose A and B are two compact operators in $\mathcal{W} \otimes \mathcal{W}'$, where $\mathcal{W}$ and $\mathcal{W}'$ are two separable Hilbert spaces. Let $\{\sigma_k(A)\}_{k=1}^\infty$ be the singular values of $\mathcal{A}$ in the decreasing order, and $\{\sigma_k(\mathcal{B})\}_{k=1}^\infty$ be the singular values of $\mathcal{B}$ in the decreasing order. Then*

$$\sum_{k=1}^{\infty} (\sigma_k(A) - \sigma_k(B))^2 \le \|A - B\|_F^2 = \sum_{k=1}^{\infty} \sigma_k^2(A-B).$$

Proof. Let $\{\phi_i\}_{i=1}^\infty$ and $\{\phi_i'\}_{i=1}^\infty$ be the orthogonal basis of $\mathcal{W}$ and $\mathcal{W}'$. Let

$$\mathcal{W}_j = \mathrm{Span}(\{\phi_i\}_{i=1}^j) \quad \text{and} \quad \mathcal{W}_j' = \mathrm{Span}(\{\phi_i'\}_{i=1}^j).$$

Denote

$$A_j = A \times \mathcal{P}_{\mathcal{W}_j} \times \mathcal{P}_{\mathcal{W}_j'} \quad \text{and} \quad B_j = B \times \mathcal{P}_{\mathcal{W}_j} \times \mathcal{P}_{\mathcal{W}_j'}.$$

Since both A and B are compact, let n be sufficiently large such that for all $j \ge n$,

$$\|A - A_j\|_F \le \epsilon \quad \text{and} \quad \|B - B_j\|_F \le \epsilon.$$

Then

$$\sqrt{\sum_{k=1}^{\infty}(\sigma_k(A)-\sigma_k(B))^2}=\sqrt{\sum_{k=1}^{\infty}(\sigma_k(A)-\sigma_k(A_j)+\sigma_k(A_j)-\sigma_k(B_j)+\sigma_k(B_j)-\sigma_k(B))^2}$$
$$\leq\sqrt{\sum_{k=1}^{\infty}(\sigma_k(A)-\sigma_k(A_j))^2}+\sqrt{\sum_{k=1}^{\infty}(\sigma_k(A_j)-\sigma_k(B_j))^2}+\sqrt{\sum_{k=1}^{\infty}(\sigma_k(B_j)-\sigma_k(B))^2}.$$

Here

$$\sqrt{\sum_{k=1}^{\infty}(\sigma_k(A)-\sigma_k(A_j))^2}=\|A-A_j\|_F\leq\epsilon \quad \text{and} \quad \sqrt{\sum_{k=1}^{\infty}(\sigma_k(B_j)-\sigma_k(B))^2}=\|B-B_j\|_F\leq\epsilon.$$

In addition, both A_j and B_j can be viewed as finite-dimensional matrices of size $j\times j$. So for $k>j$,

$$\sigma_k(A_j)=\sigma_k(B_j)=0.$$

By finite-dimensional Mirsky in Theorem 17,

$$\sqrt{\sum_{k=1}^{\infty}(\sigma_k(A_j)-\sigma_k(B_j))^2}=\sqrt{\sum_{k=1}^{j}(\sigma_k(A_j)-\sigma_k(B_j))^2}\leq\sqrt{\sum_{k=1}^{j}(\sigma_k(A_j-B_j))^2}$$
$$=\|A_j-B_j\|_F\leq\|A_j-A\|_F+\|A-B\|_F+\|B-B_j\|_F\leq 2\epsilon+\|A-B\|_F.$$

Therefore

$$\sqrt{\sum_{k=1}^{\infty}(\sigma_k(A)-\sigma_k(B))^2}\leq 4\epsilon+\|A-B\|_F.$$

The desired result follows because ϵ is arbitrary. □

A.5 Approximation Theories

In this section, we justify Theorem 15.

Reproducing Kernel Hilbert Space Basis

Let $\mathcal{O}$ be a measurable set in $\mathbb{R}$. For $x, y \in \mathcal{O}$, let $\mathbb{K} : \mathcal{O} \times \mathcal{O} \to \mathbb{R}$ be a kernel function such that

$$\mathbb{K}(x, y) = \sum_{k=1}^{\infty} \lambda_k^{\mathbb{K}} \phi_k^{\mathbb{K}}(x) \phi_k^{\mathbb{K}}(y), \tag{A.72}$$

where $\{\lambda_k^{\mathbb{K}}\}_{k=1}^{\infty} \subset \mathbb{R}^+ \cup \{0\}$, and $\{\phi_k^{\mathbb{K}}\}_{k=1}^{\infty}$ is a collection of basis functions in $L_2(\mathcal{O})$.

The reproducing kernel Hilbert space generated by $\mathbb{K}$ is

$$\mathcal{H}(\mathbb{K}) = \left\{ f \in L_2([0,1]) : \|f\|_{\mathcal{H}(\mathbb{K})}^2 = \sum_{k=1}^{\infty} (\lambda_k^{\mathbb{K}})^{-1} \langle f, \phi_k^{\mathbb{K}} \rangle^2 < \infty \right\}. \tag{A.73}$$

For any functions $f, g \in \mathcal{H}(\mathbb{K})$, the inner product in $\mathcal{H}(\mathbb{K})$ is given by

$$\langle f, g \rangle_{\mathcal{H}(\mathbb{K})} = \sum_{k=1}^{\infty} (\lambda_k^{\mathbb{K}})^{-1} \langle f, \phi_k^{\mathbb{K}} \rangle \langle g, \phi_k^{\mathbb{K}} \rangle.$$

Denote $\Theta_k^{\mathbb{K}} = (\lambda_k^{\mathbb{K}})^{-1/2} \phi_k^{\mathbb{K}}$. Then $\{\Theta_k^{\mathbb{K}}\}_{k=1}^{\infty}$ are the orthonormal basis functions in $\mathcal{H}(\mathbb{K})$ as we have that

$$\langle \Theta_{k_1}^{\mathbb{K}}, \Theta_{k_2}^{\mathbb{K}} \rangle_{\mathcal{H}(\mathbb{K})} = \begin{cases} 1, & \text{if } k_1 = k_2; \\ 0, & \text{if } k_1 \neq k_2. \end{cases}$$

an that

$$\|f\|_{\mathcal{H}(\mathbb{K})}^2 = \sum_{k=1}^{\infty} (\lambda_k^{\mathbb{K}})^{-1} \langle f, \phi_k^{\mathbb{K}} \rangle^2 = \sum_{k=1}^{\infty} \langle f, \Theta_k^{\mathbb{K}} \rangle^2.$$

Define the tensor product space

$$\underbrace{\mathcal{H}(\mathbb{K}) \otimes \cdots \otimes \mathcal{H}(\mathbb{K})}_{d \text{ copies}} = \mathcal{H}(\mathbb{K}^{\otimes d}).$$

The induced Frobenius norm in $\mathcal{H}(\mathbb{K}^{\otimes d})$ is

$$\|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d})} = \sum_{k_1,\ldots,k_d=1}^{\infty} \langle A, \Theta^{\mathbb{K}}_{k_1} \otimes \cdots \otimes \Theta^{\mathbb{K}}_{k_d} \rangle^2 = \sum_{k_1,\ldots,k_d=1}^{\infty} (\lambda^{\mathbb{K}}_{k_1} \ldots \lambda^{\mathbb{K}}_{k_d})^{-1} \langle A, \phi^{\mathbb{K}}_{k_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{k_d} \rangle^2, \tag{A.74}$$

where $\langle A, \phi^{\mathbb{K}}_{k_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{k_d} \rangle$ is defined in (A.3). The following lemma shows that the space $\mathcal{H}(\mathbb{K}^{\otimes d})$ is naturally motivated by multidimensional Sobolev spaces.

Lemma 25. *Let $\mathcal{O} = [0,1]$. With $\lambda^{\mathbb{K}}_k \asymp k^{-2\alpha}$ and suitable choices of $\{\phi^{\mathbb{K}}_k\}_{k=1}^{\infty}$, it holds that*

$$\mathcal{H}(\mathbb{K}^{\otimes d}) = W_2^{\alpha}([0,1]^d).$$

Proof. Let $\mathcal{O} = [0,1]$. When $d = 1$, it is a classical Functional analysis result that with $\lambda^{\mathbb{K}}_k \asymp k^{-2\alpha}$ and suitable choices of $\{\phi^{\mathbb{K}}_k\}_{k=1}^{\infty}$,

$$\mathcal{H}(\mathbb{K}) = W_2^{\alpha}([0,1]).$$

We refer interested readers to Chapter 12 of [223] for more details. In general, it is well-known in functional analysis that for $\Omega_1 \subset \mathbb{R}^{d_1}$ and $\Omega_2 \subset \mathbb{R}^{d_2}$, then

$$W_2^{\alpha}(\Omega_1) \otimes W_2^{\alpha}(\Omega_2) = W_2^{\alpha}(\Omega_1 \times \Omega_2).$$

Therefore by induction

$$\mathcal{H}(\mathbb{K}^{\otimes d}) = \mathcal{H}(\mathbb{K}^{\otimes d-1}) \otimes \mathcal{H}(\mathbb{K}) = W_2^{\alpha}([0,1]^{d-1}) \otimes W_2^{\alpha}([0,1]) = W_2^{\alpha}([0,1]^d).$$

□

Let $d_1 + d_2 = d$. In what follows, we show Assumption 8 holds when

$$\mathcal{M} = \operatorname{span}\left\{\phi^{\mathbb{K}}_{l_1}(z_1)\cdots\phi^{\mathbb{K}}_{l_{d_1}}(z_{d_1})\right\}^{m}_{l_1,\ldots,l_{d_1}=1} \quad \text{and} \quad \mathcal{N} = \operatorname{span}\left\{\phi^{\mathbb{K}}_{\eta_1}(z_{d_1+1})\cdots\phi^{\mathbb{K}}_{\eta_{d_2}}(z_d)\right\}^{\ell}_{\eta_1,\ldots,\eta_{d_2}=1}.$$

Lemma 26. *Let $\mathbb{K}$ be a kernel in the form of* (A.72)*. Suppose that $\lambda^{\mathbb{K}}_k \asymp k^{-2\alpha}$, and that $A : \mathcal{O}^d \to \mathbb{R}$ is such that $\|A\|_{\{\mathcal{H}(\mathbb{K})\}^{\otimes d}} < \infty$. Then for any two positive integers $m, \ell \in \mathbb{Z}^+$, it holds that*

$$\|A - A\times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{N}}\|^2_{L_2(\mathcal{O}^d)} \le C(m^{-2\alpha} + \ell^{-2\alpha})\|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d})}, \tag{A.75}$$

where C is some absolute constant. Consequently

$$\|A - A\times_y \mathcal{P}_{\mathcal{N}}\|^2_{L_2(\mathcal{O}^d)} \le C\ell^{-2\alpha}\|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d})} \quad \textit{and} \tag{A.76}$$

$$\|A\times_y \mathcal{P}_{\mathcal{N}} - A\times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{N}}\|^2_{L_2(\mathcal{O}^d)} \le Cm^{-2\alpha}\|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d})}. \tag{A.77}$$

Proof. Since $\lambda^{\mathbb{K}}_k \asymp k^{-2\alpha}$, without loss of generality, throughout the proof we assume that

$$\lambda^{\mathbb{K}}_k = k^{-2\alpha}, \tag{A.78}$$

as otherwise all of our analysis still holds up to an absolute constant. Observe that

$$\begin{aligned}
&\langle A\times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{N}}\rangle), (\phi^{\mathbb{K}}_{l_1}\otimes\cdots\otimes\phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1}\otimes\cdots\otimes\phi^{\mathbb{K}}_{\eta_{d_2}})\rangle \\
&= \begin{cases} \langle A, (\phi^{\mathbb{K}}_{l_1}\otimes\cdots\otimes\phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1}\otimes\cdots\otimes\phi^{\mathbb{K}}_{\eta_{d_2}})\rangle, & \text{if } 1 \le l_1,\ldots,l_{d_1} \le m \text{ and } 1 \le \eta_1,\ldots,\eta_{d_2} \le \ell, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

Then

$$\|A - A\times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{N}}\|^2_{L_2(\mathcal{O}^d)} = \sum_{l_1=m+1}^{\infty} \sum_{l_2,\ldots,l_{d_1}=1}^{m} \sum_{\eta_1,\ldots,\eta_{d_2}=1}^{\ell} \langle A, (\phi^{\mathbb{K}}_{l_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{\eta_{d_2}})\rangle^2$$

$$+ \ldots$$

$$+ \sum_{l_1,\ldots,l_{d_1}=1}^{m} \sum_{\eta_1,\ldots,\eta_{d_2-1}=1}^{\ell} \sum_{\eta_{d_2}=\ell+1}^{\infty} \langle A, (\phi^{\mathbb{K}}_{l_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{\eta_{d_2}})\rangle^2$$

Observe that

$$\sum_{l_1=m+1}^{\infty} \sum_{l_2,\ldots,l_{d_1}=1}^{m} \sum_{\eta_1,\ldots,\eta_{d_2}=1}^{\ell} \langle A, (\phi^{\mathbb{K}}_{l_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{\eta_{d_2}})\rangle^2$$

$$\leq \sum_{l_1=m+1}^{\infty} m^{-2\alpha} l_1^{2\alpha} \sum_{l_2,\ldots,l_{d_1}=1}^{m} \sum_{\eta_1,\ldots,\eta_{d_2}=1}^{\ell} \langle A, (\phi^{\mathbb{K}}_{l_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{\eta_{d_2}})\rangle^2$$

$$\leq m^{-2\alpha} \sum_{l_1=m+1}^{\infty} \sum_{l_2,\ldots,l_{d_1}=1}^{m} \sum_{\eta_1,\ldots,\eta_{d_2}=1}^{\ell} l_1^{2\alpha} \cdots l_{d_1}^{2\alpha} \eta_1^{2\alpha} \cdots \eta_{d_2}^{2\alpha} \langle A, (\phi^{\mathbb{K}}_{l_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{\eta_{d_2}})\rangle^2$$

$$\leq m^{-2\alpha} \sum_{l_1=1}^{\infty} \sum_{l_2,\ldots,l_{d_1}=1}^{\infty} \sum_{\eta_1,\ldots,\eta_{d_2}=1}^{\infty} l_1^{2\alpha} \cdots l_{d_1}^{2\alpha} \eta_1^{2\alpha} \cdots \eta_{d_2}^{2\alpha} \langle A, (\phi^{\mathbb{K}}_{l_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{l_{d_1}}) \otimes (\phi^{\mathbb{K}}_{\eta_1} \otimes \cdots \otimes \phi^{\mathbb{K}}_{\eta_{d_2}})\rangle^2$$

$$= m^{-2\alpha} \|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d-1})}$$

where the first inequality holds because $l_1 \geq m+1 \geq m$ and the last equality follows from

(A.74) and (A.78). Similarly

$$\begin{aligned}
&\sum_{l_1,\ldots,l_{d_1}=1}^{m}\sum_{\eta_1,\ldots,\eta_{d_2-1}=1}^{\ell}\sum_{\eta_{d_2}=\ell+1}^{\infty}\langle A,(\phi_{l_1}^{\mathbb{K}}\otimes\cdots\otimes\phi_{l_{d_1}}^{\mathbb{K}})\otimes(\phi_{\eta_1}^{\mathbb{K}}\otimes\cdots\otimes\phi_{\eta_{d_2}}^{\mathbb{K}})\rangle^2\\
\leq&\sum_{l_1,\ldots,l_{d_1}=1}^{m}\sum_{\eta_1,\ldots,\eta_{d_2-1}=1}^{\ell}\sum_{\eta_{d_2}=\ell+1}^{\infty}\ell^{-2\alpha}\eta_{d_2}^{2\alpha}\langle A,(\phi_{l_1}^{\mathbb{K}}\otimes\cdots\otimes\phi_{l_{d_1}}^{\mathbb{K}})\otimes(\phi_{\eta_1}^{\mathbb{K}}\otimes\cdots\otimes\phi_{\eta_{d_2}}^{\mathbb{K}})\rangle^2\\
\leq&\ell^{-2\alpha}\sum_{l_1,\ldots,l_{d_1}=1}^{m}\sum_{\eta_1,\ldots,\eta_{d_2-1}=1}^{\ell}\sum_{\eta_{d_2}=\ell+1}^{\infty}l_1^{2\alpha}\cdots l_{d_1}^{2\alpha}\eta_1^{2\alpha}\cdots\eta_{d_2}^{2\alpha}\langle A,(\phi_{l_1}^{\mathbb{K}}\otimes\cdots\otimes\phi_{l_{d_1}}^{\mathbb{K}})\otimes(\phi_{\eta_1}^{\mathbb{K}}\otimes\cdots\otimes\phi_{\eta_{d_2}}^{\mathbb{K}})\rangle^2\\
\leq&\ell^{-2\alpha}\|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d-1})},
\end{aligned}$$

where the first inequality holds because $\eta_{d_2}\geq\ell+1\geq\ell$ and the last inequality follows from (A.74) and (A.78). Thus (A.75) follows immediately.

For (A.76), note that when $m=\infty$, $\mathcal{M}=L_2(\mathcal{O}^{d_1})$. In this case $\mathcal{P}_{\mathcal{M}}$ becomes the identity operator and

$$A\times_x\mathcal{P}_{\mathcal{M}}\times_y\mathcal{P}_{\mathcal{N}}=A\times_y\mathcal{P}_{\mathcal{N}}.$$

Therefore (A.76) follows from (A.75) by taking $m=\infty$.

For (A.77), similar to (A.76), we have that

$$\|A-A\times_x\mathcal{P}_{\mathcal{M}}\|^2_{L_2(\mathcal{O}^d)}\leq Cm^{-2\alpha}\|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d})}.$$

It follows that

$$\|A\times_y\mathcal{P}_{\mathcal{N}}-A\times_x\mathcal{P}_{\mathcal{M}}\times_y\mathcal{P}_{\mathcal{N}}\|^2_{L_2(\mathcal{O}^d)}\leq\|A-A\times_x\mathcal{P}_{\mathcal{M}}\|^2_{L_2(\mathcal{O}^d)}\|\mathcal{P}_{\mathcal{N}}\|_2^2\leq Cd_1m^{-2\alpha}\|A\|^2_{\mathcal{H}(\mathbb{K}^{\otimes d})},$$

where last inequality follows from the fact that $\|\mathcal{P}_{\mathcal{N}}\|_2\leq 1$. □

Legendre Polynomial Basis

Legendre polynomials is a well-known classical orthonormal polynomial system in $L_2([-1,1])$. We can define the Legendre polynomials in the following inductive way. Let $p_0 = 1$ and suppose $\{p_k\}_{k=1}^{n-1}$ are defined. Let $p_n : [-1,1] \to \mathbb{R}$ be a polynomial of degree n such that

- $\|p_n\|_{L_2([-1,1])} = 1$, and
- $\int_{-1}^{1} p_n(z)p_k(z)\mathrm{d}z = 0$ for all $0 \le k \le n-1$.

As a quick example, we have that

$$p_0(z) = 1, \quad p_1(z) = \sqrt{\frac{3}{2}}z, \quad \text{and} \quad p_2(z) = \sqrt{\frac{5}{3}}\frac{3z^2-1}{2}.$$

Let $q_k(z) = \sqrt{2}p_k(2z-1)$. Then $\{q_k\}_{k=0}^{\infty}$ are the orthonormal polynomial system in $L_2([0,1])$. In this subsection, we show that Assumption 8 holds when $\{\phi_k\}_{k=1}^{\infty} = \{q_k\}_{k=0}^{\infty}$. More precisely, let

$$\mathbb{S}_n(z) = \mathrm{Span}\{q_k(z)\}_{k=0}^{n}$$

and $\mathcal{P}_{\mathbb{S}_n(z)}$ denote the projection operator from $L_2([0,1])$ to $\mathbb{S}_n(z)$. Then $\mathbb{S}_n(z)$ is the subspace of polynomials of degree at most n. In addition, for any $f \in L_2([0,1])$, $\mathcal{P}_{\mathbb{S}_n(z)}(f)$ is the best n-degree polynomial approximation of f in the sense that

$$\|\mathcal{P}_{\mathbb{S}_n(z)}(f) - f\|_{L_2([0,1])} = \min_{g\in\mathbb{S}_n} \|g - f\|_{L_2([0,1])}. \tag{A.79}$$

We begin with a well-known polynomial approximation result. For $\alpha \in \mathbb{Z}^+$, denote $C^{\alpha}([0,1])$ to be the class of functions that are α times continuously differentiable.

Theorem 19. *Suppose $f \in C^{\alpha}([0,1])$. Then for any $n \in \mathbb{Z}^+$, there exists a polynomial $g_{2n}(f)$ of degree $2n$ such that*

$$\|f - g_{2n}(f)\|_{L_2([0,1])}^2 \le Cn^{-2\alpha}\|f^{(\alpha)}\|_{L_2([0,1])}^2,$$

where C is an absolute constant.

Proof. This is Theorem 1.2 of [237]. □

Therefore by (A.79) and Theorem 19,

$$\|\mathcal{P}_{\mathbb{S}_n(z)}(f) - f\|^2_{L_2([0,1])} \le \|f - g_{\lfloor n/2 \rfloor}(f)\|^2_{L_2([0,1])} \le C' n^{-2\alpha} \|f^{(\alpha)}\|^2_{L_2([0,1])}. \tag{A.80}$$

Let $z = (z_1, \ldots, z_d) \in \mathbb{R}^d$ and let $\mathbb{S}_n(z_j)$ denote the linear space spanned by polynomials of z_j of degree at most n.

Corollary 6. *Suppose $B(z_1, \ldots, z_d) \in C^\alpha([0,1]^d)$. Then for any $1 \le k \le d$,*

$$\|B - B \times_k \mathcal{P}_{\mathbb{S}_n(z_k)}\|_{L_2([0,1]^d)} \le C n^{-\alpha} \|B\|_{W_2^\alpha([0,1]^d)}.$$

Proof. It suffices to consider $k = 1$. For any fixed $(z_2, \ldots, z_d)$, $B(\cdot, z_2, \ldots, z_d) \in C^\alpha([0,1])$. Therefore by (A.80),

$$\int \left\{ B(z_1, z_2, \ldots, z_d) - B \times_1 \mathcal{P}_{\mathbb{S}_n(z_1)}(z_1, z_2, \ldots, z_d) \right\}^2 \mathrm{d}z_1 \le C n^{-2\alpha} \int \left\{ \frac{\partial^\alpha}{\partial^\alpha z_1} B(z_1, z_2, \ldots, z_d) \right\}^2 \mathrm{d}z_1.$$

Therefore

$$\begin{aligned}
&\|B - B \times_1 \mathcal{P}_{\mathbb{S}_n(z_1)}\|_{L_2([0,1]^d)} \\
=& \int \cdots \int \left\{ B(z_1, z_2, \ldots, z_d) - B \times_1 \mathcal{P}_{\mathbb{S}_n(z_1)}(z_1, z_2, \ldots, z_d) \right\}^2 \mathrm{d}z_1 \mathrm{d}z_2 \cdots \mathrm{d}z_d \\
\le& \int \cdots \int C n^{-2\alpha} \int \left\{ \frac{\partial^\alpha}{\partial^\alpha z_1} B(z_1, z_2, \ldots, z_d) \right\}^2 \mathrm{d}z_1 \mathrm{d}z_2 \cdots \mathrm{d}z_d.
\end{aligned}$$

The desired result follows from the observation that $\int \cdots \int \left\{ \frac{\partial^\alpha}{\partial^\alpha z_1} B(z_1, z_2, \ldots, z_d) \right\}^2 \mathrm{d}z_1 \cdots \mathrm{d}z_d \le \|B\|^2_{W_2^\alpha([0,1]^d)}$. □

In what follows, we present a polynomial approximation theory in multidimensions.

Lemma 27. *For $j \in [1, \ldots, d]$, let $\mathbb{S}_{n_j}(z_j)$ denote the linear space spanned by polynomials of z_j of degree n_j and let $\mathcal{P}_{\mathbb{S}_{n_j}(z_j)}$ be the corresponding projection operator. Then for any $B \in W_2^{\alpha}([0,1]^d)$, it holds that*

$$\|B - B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_d \mathcal{P}_{\mathbb{S}_{n_d}(z_d)}\|_{L_2([0,1]^d)} \leq C \sum_{j=1}^{d} n_j^{-\alpha} \|B\|_{W_2^{\alpha}([0,1]^d)}. \tag{A.81}$$

Proof. Since $C^{\alpha}([0,1]^d)$ is dense in $W_2^{\alpha}([0,1]^d)$, it suffices to show (A.81) for all functions in $\in C^{\alpha}([0,1]^d)$. To this end, we proceed by induction. The base case

$$\|B - B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)}\|_{L_2([0,1]^d)} \leq C n_1^{-\alpha} \|B\|_{W_2^{\alpha}([0,1]^d)}$$

is a direct consequence of Corollary 6. Suppose by induction, the following inequality holds for k,

$$\|B - B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_k \mathcal{P}_{\mathbb{S}_{n_k}(z_k)}\|_{L_2([0,1]^d)} \leq C \sum_{j=1}^{k} n_j^{-\alpha} \|B\|_{W_2^{\alpha}([0,1]^d)}. \tag{A.82}$$

Then

$$\begin{aligned}
&\|B - B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_k \mathcal{P}_{\mathbb{S}_{n_k}(z_k)} \times_{k+1} \mathcal{P}_{\mathbb{S}_{n_{k+1}}(z_{k+1})}\|_{L_2([0,1]^d)} \\
\leq &\|B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_k \mathcal{P}_{\mathbb{S}_{n_k}(z_k)} \times_{k+1} \mathcal{P}_{\mathbb{S}_{n_{k+1}}(z_{k+1})} - B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_k \mathcal{P}_{\mathbb{S}_{n_k}(z_k)}\|_{L_2([0,1]^d)} \\
+ &\|B - B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_k \mathcal{P}_{\mathbb{S}_{n_k}(z_k)}\|_{L_2([0,1]^d)}.
\end{aligned}$$

The desired result follows from (A.82) and the observation that

$$\begin{aligned}
&\|B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_k \mathcal{P}_{\mathbb{S}_{n_k}(z_k)} \times_{k+1} \mathcal{P}_{\mathbb{S}_{n_{k+1}}(z_{k+1})} - B \times_1 \mathcal{P}_{\mathbb{S}_{n_1}(z_1)} \cdots \times_k \mathcal{P}_{\mathbb{S}_{n_k}(z_k)}\|_{L_2([0,1]^d)} \\
\leq &\|B \times_{k+1} \mathcal{P}_{\mathbb{S}_{n_{k+1}}(z_{k+1})} - B\|_{L_2([0,1]^d)} \|\mathcal{P}_{\mathbb{S}_{n_1}(z_1)}\|_2 \cdots \|\mathcal{P}_{\mathbb{S}_{n_k}(z_k)}\|_2 \\
\leq &C n_{k+1}^{-\alpha} \|B\|_{W_2^\alpha([0,1]^d)},
\end{aligned}$$

where the last inequality follows from Corollary 6 and that $\|\mathcal{P}_{\mathbb{S}_{n_j}(z_j)}\|_2 \leq 1$ for all j. □

This following lemma shows that Assumption 8 is a consequence of Lemma 27.

Lemma 28. *Let* $1 \leq d_1 \leq d$, $\mathcal{M} = \mathbb{S}_m(z_1) \otimes \cdots \otimes \mathbb{S}_m(z_{d_1})$, *and* $\mathcal{N} = \mathbb{S}_\ell(z_{d_1+1}) \otimes \cdots \otimes \mathbb{S}_\ell(z_d)$. *Suppose* $\|A\|_{W_2^\alpha([0,1]^d)} < \infty$. *Then for* $1 \leq m \leq \infty$ *and* $1 \leq \ell \leq \infty$,

$$\|A - A \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}\|^2_{L_2([0,1]^d)} = \mathrm{O}(m^{-2\alpha} + \ell^{-2\alpha}) \tag{A.83}$$

$$\|A - A \times_y \mathcal{P}_\mathcal{N}\|^2_{L_2([0,1]^d)} = \mathrm{O}(\ell^{-2\alpha}) \quad \textit{and} \tag{A.84}$$

$$\|A \times_y \mathcal{P}_\mathcal{N} - A \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}\|^2_{L_2([0,1]^d)} = \mathrm{O}(m^{-2\alpha}). \tag{A.85}$$

Proof. For (A.83), by Lemma 27,

$$\begin{aligned}
\|A - A \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N}\|_{L_2([0,1]^d)} \leq &C\big(d_1(m - \alpha - 1)^{-\alpha} + d_2(\ell - \alpha - 1)^{-\alpha}\big)\|A\|_{W_2^\alpha([0,1]^d)} \\
= &\mathrm{O}(m^{-\alpha} + \ell^{-\alpha}),
\end{aligned}$$

where the equality follows from the fact that α, d_1 and d are constants.

For (A.84), note that when $m = \infty$, $\mathcal{M} = L_2([0,1]^{d_1})$. In this case $\mathcal{P}_\mathcal{M}$ becomes the identity operator and

$$A \times_x \mathcal{P}_\mathcal{M} \times_y \mathcal{P}_\mathcal{N} = A \times_y \mathcal{P}_\mathcal{N}.$$

Therefore (A.84) follows from (A.83) by taking $m = \infty$.

For (A.85), similar to (A.84), we have that

$$\|A - A \times_x \mathcal{P}_{\mathcal{M}}\|^2_{L_2([0,1]^d)} = \mathrm{O}(m^{-2\alpha}).$$

It follows that

$$\|A \times_y \mathcal{P}_{\mathcal{N}} - A \times_x \mathcal{P}_{\mathcal{M}} \times_y \mathcal{P}_{\mathcal{N}}\|^2_{L_2([0,1]^d)} \leq \|A - A \times_x \mathcal{P}_{\mathcal{M}}\|^2_{L_2([0,1]^d)} \|\mathcal{P}_{\mathcal{N}}\|_2^2 = \mathrm{O}(m^{-2\alpha}),$$

where last inequality follows from the fact that $\|\mathcal{P}_{\mathcal{N}}\|_2 \leq 1$. □

A.6 Additional Numerical Details

Neural Network Density Estimator

For neural network estimators, we use two popular density estimation architectures: Masked Autoregressive Flow (MAF) ([170]) and Neural Autoregressive Flows (NAF) ([99]) for comparisons. Both neural networks are trained using the Adam optimizer ([115]). For MAF, we use 5 transforms and each transform is a 3-hidden layer neural network with width 128. For NAF, we choose 3 transforms and each transform is a 3-hidden layer neural network with width 128.

Kernel Methods

In our simulated experiments and real data examples, we choose Gaussian kernel for the kernel density estimators and Nadaraya–Watson kernel regression (NWKR) estimators. The bandwidths in all the numerical examples are chosen using cross-validations. We refer interested readers to [229] for an introduction to nonparametric statistics.

APPENDIX B

APPENDIX OF TENSOR DENSITY ESTIMATOR

B.1 Preliminaries

Before the proof, we introduce some definitions and notations that will be used in the main proof in Section B.2.1.

To begin with, we use $c^* = \Phi^T p^* \in \mathbb{R}^{n^d}$ to present the coefficient tensor of ground truth under a set of orthonormal basis functions $\{\phi_l^j\}_{l=1}^n$, whose entries are given by

$$c^*(l_1, \cdots, l_d) = \langle p^*, \phi_{l_1}^1 \cdots \phi_{l_d}^d \rangle.$$

We point out that for p^* with 1-cluster assumption as stated in Assumption 4, it holds that $c^* = \mathrm{diag}(w_K)\Phi^T p^*$ for any $K \geqslant 1$. In the rest of this appendix, we omit the superscripts on the basis functions unless they are needed to indicate the corresponding coordinate of dimensions. Based on the additive formulation of ground truth p^* in (3.38) and orthogonality of $\{\phi_l^j\}_{l=1}^n$, c^* can be expressed as

$$c^* = \mathbf{1}^d + c_1^* \otimes \mathbf{1}^{d-1} + \mathbf{1} \otimes c_2^* \otimes \mathbf{1}^{d-2} + \cdots + \mathbf{1}^{d-1} \otimes c_d^*, \tag{B.1}$$

where $\mathbf{1}^k \in \mathbb{R}^{n^k}$ represents the k times Kronecker product of $\mathbf{1} = [1, 0, \cdots, 0]^T \in \mathbb{R}^n$. $c_j^* \in \mathbb{R}^n$ is a vector whose l-th coordinate is given by

$$c_j^*(l) = \int_0^1 \cdots \int_0^1 p^*(x_1, \cdots, x_d)\phi_l(x_j)\mathrm{d}x_1 \cdots \mathrm{d}x_d. \tag{B.2}$$

We can further simplify the expression of c_j^* as follows. By the normalization condition of p^*, q_j^* in the expression of p^* satisfies $\sum_{j=1}^d \int_0^1 q_j^*(x_j)dx_j = 0$. Without loss of generality,

we assume that

$$\int_0^1 q_j^*(x_j)\mathrm{d}x_j = 0. \tag{B.3}$$

This further gives us

$$c_j^*(l) = \begin{cases} 0, & \text{if } l = 1; \\ \int_0^1 q_j^*(x_j)\phi_l(x_j)\mathrm{d}x_j, & \text{if } 2 \leqslant l \leqslant n. \end{cases} \tag{B.4}$$

Definition 4. *Let $\mathcal{B}_{s,d}$ be the s-cluster basis functions defined in* (3.5)*. Define the s-cluster estimator*

$$\hat{p}_s(x) := 1 + \sum_{\varphi_1 \in \mathcal{B}_{1,d}} \langle \hat{p}, \varphi_1 \rangle \varphi_1(x) + \cdots + \sum_{\varphi_s \in \mathcal{B}_{s,d}} \langle \hat{p}, \varphi_s \rangle \varphi_s(x) \tag{B.5}$$

The corresponding coefficient tensor $\hat{c}[s] \in \mathbb{R}^{n^d}$ of $\hat{p}_s$ is

$$\hat{c}[s](l_1, \cdots, l_d) = \langle \hat{p}_s, \varphi_{l_1} \cdots \varphi_{l_d} \rangle. \tag{B.6}$$

In this case, $\hat{c}[s]$ will be referred to as the s-cluster coefficient estimator of c^*.

Definition 5. *Given matrix $A \in \mathbb{R}^{m \times n}$ and positive integer $r \leqslant \min\{m, n\}$, we define function* $\mathrm{SVD} : \mathbb{R}^{m \times n} \times \mathbb{Z}^+ \to \mathbb{O}^{m \times r}$ *satisfying*

$$\mathrm{SVD}(A, r) = U \tag{B.7}$$

where $A = U\Sigma V^T$ is the top-r SVD of the matrix A.

B.2 Proof of the Main Theorems

B.2.1 Proof of Theorem 4

In this proof, we simply denote $\hat{c}[K]$ as defined in (B.6) by $\hat{c}$.

Proof. Let $\hat{G}_{1:d}$ be the TT cores of $\tilde{c}$, to bound the error $\|c^* - \hat{G}_{1:d}\|_F$, we first use Lemma 31 to formulate it as

$$\begin{aligned}
&\|c^* - \hat{G}_{1:d}\|_F^2 \\
=&\|\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)} - c^{*(d-1)}\|_F^2 &\text{(B.8)}\\
+&\|\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T \hat{c}^{(d-1)} - \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)}\|_F^2. &\text{(B.9)}
\end{aligned}$$

Later we will bound the above two terms by **Step 1** and **Step 2** respectively. Before elaborating these two steps, let us introduce two error estimates that will be used frequently in the proof of this theorem. With high probability, we have

$$\max_{k\in\{1,\cdots,d\}} \|(U_{k-1}^{*T} \otimes I_n)(c^{*(k)} - \hat{c}^{(k)})V_k^*\|_F \leqslant C_1\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N}}, \tag{B.10}$$

$$\|c^* - \hat{c}\|_F \leqslant C_2\left(\sqrt{\frac{d^K n^{2K}}{N}}\right). \tag{B.11}$$

Their proofs are provided respectively by Lemma 38 and Lemma 39. In the above inequality, C_1, C_2 are two sufficiently large universal constants. U_k^* and V_k^* are orthogonal matrices such that $U_k^* U_k^{*T} \in \mathbb{R}^{n^k \times n^k}$ is the projection matrix onto the space $\mathrm{Col}(c^{*(k)})$, and $V_k^* V_k^{*T} \in \mathbb{R}^{n^{d-k}\times n^{d-k}}$ is the projection matrix onto the space $\mathrm{Row}(c^{*(k)})$. Due to the additive formulation (B.1) of c^*, U_k^* and V_k^* have analytical expressions, shown in Lemma 35.

(B.11) provides an upper bound of the gap between empirical and ground truth coefficient tensor, which scales as $\mathrm{O}(d^{K/2})$ upon using K-cluster basis for the density estimator. Notably, with the projection U_k^* and V_k^*, this gap grows only at $\mathrm{O}(\sqrt{d\log(dn)})$ under the

Assumption 4 for $\|p^*\|_\infty$ as stated in (B.10). This is due to the 1-cluster structure of c^*, which is the key to make our error analysis sharp.

Step 1. To bound (B.8), we show by induction that for $k \in \{1, \cdots, d-1\}$,

$$\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 \leqslant C_3 \frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{N} \sum_{j=1}^{k} \frac{1}{\sigma_2^2(c^{*(j)})} \tag{B.12}$$

for a universal sufficiently large constant C_3 with high probability, where $\sigma_2(c^{*(j)})$ is the second largest singular value of the unfolding matrix $c^{*(j)}$.

We prove $k = 1$ first:

$$\|\hat{G}_1\hat{G}_1^T c^{*(1)} - c^{*(1)}\|_F^2 = \|\hat{G}_1\hat{G}_1^T c^{*(1)} - U_1^* U_1^{*T} c^{*(1)}\|_F^2 \leqslant \|\hat{G}_1\hat{G}_1^T - U_1^* U_1^{*T}\|_2^2 \|c^*\|_F^2 \tag{B.13}$$

We use the matrix perturbation theory in Corollary 8 to bound $\|\hat{G}_1\hat{G}_1^T - U_1^* U_1^{*T}\|_2$. The spectrum gap assumption in Corollary 8 holds since $\|\hat{c}^{(1)} - c^{*(1)}\|_F \leqslant O\left(\sqrt{\frac{d^K n^{2K}}{N}}\right)$ and $O\left(\sqrt{\frac{d^K n^{2K}}{N}}\right) \leqslant \sigma_2(c^{*(1)})$ by (B.11) and using a sufficiently large N such that (3.39) holds. Therefore, we have

$$\|\hat{G}_1\hat{G}_1^T - U_1^* U_1^{*T}\|_2 \leqslant C_4\left(\frac{\|(\hat{c}^{(1)} - c^{*(1)})V_1^{*T}\|_2}{\sigma_2(c^{*(1)})} + \frac{\|\hat{c}^{(1)} - c^{*(1)}\|_2^2}{\sigma_2^2(c^{*(1)})}\right). \tag{B.14}$$

for some universal constant C_4. According to (B.10), the first term

$$\frac{\|(\hat{c}^{(1)} - c^{*(1)})V_1^{*T}\|_2}{\sigma_2(c^{*(1)})} \leqslant C_1\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N\sigma_2^2(c^{*(1)})}} \tag{B.15}$$

and the second term can be bounded by the first term upon using sufficiently large N such

that (3.39) holds. Therefore,

$$\|\hat{G}_1\hat{G}_1^T c^{*(1)} - c^{*(1)}\|_F^2 \leqslant C_3 \frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{N\sigma_2^2(c^{*(1)})} \tag{B.16}$$

for some universal constant C_3.

Now suppose (B.12) holds for k. Then for $k+1$,

$$\begin{aligned}
&\|\hat{G}_{1:k+1}\hat{G}_{1:k+1}^T c^{*(k+1)} - c^{*(k+1)}\|_F^2 \\
=&\|\hat{G}_{1:k+1}\hat{G}_{1:k+1}^T c^{*(k+1)} - (\hat{G}_{1:k}\otimes I_n)(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}\|_F^2 + \|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 \\
\leqslant&\|\hat{G}_{1:k+1}\hat{G}_{1:k+1}^T c^{*(k+1)} - (\hat{G}_{1:k}\otimes I_n)(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}\|_F^2 \qquad \text{(B.17)} \\
&+C_3 \frac{n\log(dn)\|p^*\|_\infty\|c^*\|_F^2}{N}\sum_{j=1}^{k}\frac{1}{\sigma_2^2(c^{*(j)})}. \qquad \text{(B.18)}
\end{aligned}$$

where the equality follows from Lemma 30, and the inequality follows from induction (B.12). It suffices to bound (B.17). Note that

$$\begin{aligned}
\text{(B.17)} =&\|(\hat{G}_{1:k}\otimes I_n)\hat{G}_{k+1}\hat{G}_{k+1}^T(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)} - (\hat{G}_{1:k}\otimes I_n)(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}\|_F^2 \\
=&\|(\hat{G}_{1:k}\otimes I_n)\big[\hat{G}_{k+1}\hat{G}_{k+1}^T(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)} - (\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}\big]\|_F^2 \\
\leqslant&\|\hat{G}_{1:k}\otimes I_n\|_2^2\|\hat{G}_{k+1}\hat{G}_{k+1}^T(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)} - (\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}\|_F^2 \\
=&\|\hat{G}_{k+1}\hat{G}_{k+1}^T(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)} - (\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}\|_F^2. \qquad \text{(B.19)}
\end{aligned}$$

Step 1A. Define the matrix formed by principal left singular vectors:

$$\tilde{G}_{k+1} = \text{SVD}\left((\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}, 2\right) \in \mathbb{O}^{2n\times 2}. \tag{B.20}$$

where the SVD function is defined in (B.7). Since $\text{Rank}((\hat{G}_{1:k-1}\otimes I_n)^T c^{*(k)}) = 2$, it follows

that

$$(\hat{G}_{1:k} \otimes I_n)^T c^{*(k+1)} = \tilde{G}_{k+1}\tilde{G}_{k+1}^T(\hat{G}_{1:k} \otimes I_n)^T c^{*(k+1)}. \tag{B.21}$$

Therefore by (B.19) and (B.21),

$$\begin{aligned}
\text{(B.17)} =& \|\hat{G}_{k+1}\hat{G}_{k+1}^T(\hat{G}_{1:k+1} \otimes I_n)^T c^{*(k+1)} - \tilde{G}_{k+1}\tilde{G}_{k+1}^T(\hat{G}_{1:k} \otimes I_n)^T c^{*(k+1)}\|_F^2 \\
\leqslant& \|\hat{G}_{k+1}\hat{G}_{k+1}^T - \tilde{G}_{k+1}\tilde{G}_{k+1}^T\|_2^2 \|\hat{G}_{1:k} \otimes I_n\|_2^2 \|c^{*(k+1)}\|_F^2 \\
=& \|\hat{G}_{k+1}\hat{G}_{k+1}^T - \tilde{G}_{k+1}\tilde{G}_{k+1}^T\|_2^2 \|c^*\|_F^2.
\end{aligned} \tag{B.22}$$

Perturbation of projection matrices $\|\hat{G}_{k+1}\hat{G}_{k+1}^T - \tilde{G}_{k+1}\tilde{G}_{k+1}^T\|_2$ will again be estimated using Corollary 8. More specifically, in Corollary 8, we consider $\hat{A} = (\hat{G}_{1:k} \otimes I_n)^T \hat{c}^{(k+1)}$ and $A = \hat{G}_{1:k} \otimes I_n)^T c^{*(k+1)}$, and $\hat{G}_{k+1}\hat{G}_{k+1}^T$ and $\tilde{G}_{k+1}\tilde{G}_{k+1}^T$ are respectively the projection matrices of $\hat{A}$ and A. In the subsequent **Step 1B**, we verify the conditions that should be satisfied in Corollary 8.

Step 1B. By the induction assumption (B.12), we have

$$\begin{aligned}
\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 &\leqslant C_3 \frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{N} \sum_{j=1}^{k} \frac{1}{\sigma_2^2(c^{*(j)})} \\
&\leqslant \frac{C_3 n\log(dn)(1+B^*d)C_c d}{N} \sum_{j=1}^{d} \frac{d}{c_\sigma^2 j(d-j)} \leqslant \left(\frac{B^*}{b^*}\right)^4 \frac{C_4 n d^2 \log^2(dn)}{N},
\end{aligned} \tag{B.23}$$

where C_4 is a universal constant, B^* is given in Assumption 4. The second inequality above follows from upper bound for $\|c^*\|_F^2$ in Corollary 7 and the lower bound for singular value in Lemma 37, and the third inequality follows from the fact that $\sum_{j=1}^{d} \frac{1}{j(d-j)} = \mathrm{O}(\frac{\log(d)}{d})$.

Consequently,

$$\|(\hat{G}_{1:k}\otimes I_n)(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)} - c^{*(k+1)}\|_F^2 = \|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 \leqslant \frac{\sigma_2^2(c^{*(k+1)})}{4}, \tag{B.24}$$

where the inequality follows from the lower bound for the singular value $c^{*(k)}$ given in Lemma 37 and the assumption (3.39) on sample size N. Therefore, by Lemma 45,

- $\text{Rank}((\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}) = 2$; (B.25)
- $V_{k+1}^* V_{k+1}^{*T}$ is the projection matrix on to $\text{Row}((\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)})$; (B.26)
- $\sigma_2((\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)}) \geqslant \frac{\sigma_2(c^{*(k+1)})}{2}$. (B.27)

We start to apply Corollary 8 to $\hat{G}_{k+1}$ and $\tilde{G}_{k+1}$, observe that

$$\begin{aligned}&\|(\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)} - (\hat{G}_{1:k}\otimes I_n)^T \hat{c}^{(k+1)}\|_F \leqslant \|\hat{G}_{1:k}\otimes I_n\|_2 \|c^{*(k+1)} - \hat{c}^{(k+1)}\|_F \\ =&\|\hat{c} - c^*\|_F \leqslant C_2\left(\sqrt{\frac{d^K n^{2K}}{N}}\right) \leqslant 8\sigma_2(c^{*(k+1)}) \leqslant 4\sigma_2((\hat{G}_{1:k}\otimes I_n)^T c^{*(k)}),\end{aligned} \tag{B.28}$$

where the equality follows from the fact that $\hat{G}_{1:k}\otimes I_n \in \mathbb{O}^{n^{k+1}\times 2n}$, the second inequality follows from (B.11), the third inequality follows from (3.39), and the last inequality follows from (B.27). Since (B.28) and (B.26) hold, by Corollary 8

$$\begin{aligned}&\|\tilde{G}_{k+1}\tilde{G}_{k+1}^T - \hat{G}_{k+1}\hat{G}_{k+1}\|_2 \\ \leqslant&\frac{\|(\hat{G}_{1:k}\otimes I_n)^T(c^{*(k+1)} - \hat{c}^{(k+1)})V_{k+1}^*\|_2}{\sigma_2((\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)})} &\text{(B.29)}\\ +&\frac{\|(\hat{G}_{1:k}\otimes I_n)^T(c^{*(k+1)} - \hat{c}^{(k+1)})\|_2^2}{\sigma_2^2((\hat{G}_{1:k}\otimes I_n)^T c^{*(k+1)})^2} &\text{(B.30)}\end{aligned}$$

We will bound (B.29) in **Step 1C** and (B.30) in **Step 1D**.

Step 1C. For (B.29), we have

$$\begin{aligned}
&\|(\hat{G}_{1:k}\otimes I_n)^T(c^{*(k+1)}-\hat{c}^{(k+1)})V_{k+1}^*\|_2\\
=&\|(\hat{G}_{1:k}\otimes I_n)(\hat{G}_{1:k}\otimes I_n)^T(c^{*(k+1)}-\hat{c}^{(k+1)})V_{k+1}^*\|_2\\
=&\|(\hat{G}_{1:k}\hat{G}_{1:k}^T\otimes I_n)(c^{*(k+1)}-\hat{c}^{(k+1)})V_{k+1}^*\|_2\\
\leqslant&\|(\{U_k^*U_k^{*T}-\hat{G}_{1:k}\hat{G}_{1:k}^T\}\otimes I_n)(c^{*(k+1)}-\hat{c}^{(k+1)})V_{k+1}^*\|_2 \qquad \text{(B.31)}\\
+&\|(U_k^*U_k^{*T}\otimes I_n)(c^{*(k+1)}-\hat{c}^{(k+1)})V_{k+1}^*\|_2. \qquad \text{(B.32)}
\end{aligned}$$

Note that

$$\begin{aligned}
\text{(B.32)} =&\|(U_k^*\otimes I_n)(U_k^*\otimes I_n)^T(c^{*(k+1)}-\hat{c}^{(k+1)})V_{k+1}^*\|_2\\
=&\|(U_k^*\otimes I_n)^T(c^{*(k+1)}-\hat{c}^{(k+1)})V_{k+1}^*\|_2 \qquad \text{(B.33)}\\
\leqslant&C_1\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N}}\leqslant\frac{\sqrt{C_3}}{8}\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N}},
\end{aligned}$$

where the last inequality follows from (B.10) and we choose C_3 in (B.16) to be at least $64C_1^2$. In addition,

$$\begin{aligned}
\text{(B.31)} \leqslant&\|\{U_k^*U_k^{*T}-\hat{G}_{1:k}\hat{G}_{1:k}^T\}\otimes I_n\|_2\|c^{*(k+1)}-\hat{c}^{(k+1)}\|_2\|V_{k+1}^*\|_2\\
=&\|U_k^*U_k^{*T}-\hat{G}_{1:k}\hat{G}_{1:k}^T\|_2\|c^*-\hat{c}\|_2\leqslant\|U_k^*U_k^{*T}-\hat{G}_{1:k}\hat{G}_{1:k}^T\|_2C_2\left(\sqrt{\frac{d^Kn^{2K}}{N}}\right)\\
\leqslant&C_4\frac{\sqrt{nB^{*3}\|p^*\|_\infty}}{b^{*2}\sigma_2(c^{*(k)})}\sqrt{\frac{d\log(dn)}{N}}C_2\left(\sqrt{\frac{d^Kn^{2K}}{N}}\right)\leqslant\frac{\sqrt{C_3}}{8}\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N}}, \qquad \text{(B.34)}
\end{aligned}$$

where the second inequality follows from (B.11), the third inequality in which C_4 is a universal constant follows from Lemma 32, and the last inequality holds for sufficiently large N in (3.39).

Therefore (B.31), (B.32), (B.33), and (B.34) together lead to

$$\|(\hat{G}_{1:k} \otimes I_n)^T (c^{*(k+1)} - \hat{c}^{(k+1)}) V_{k+1}^*\|_2 \leqslant \frac{\sqrt{C_3}}{4} \sqrt{\frac{n \log(dn) \|p^*\|_\infty}{N}}. \tag{B.35}$$

So by (B.35) and (B.27),

$$\text{(B.29)} \leqslant \frac{\sqrt{C_3}}{2} \frac{\sqrt{n \log(dn) \|p^*\|_\infty}}{\sqrt{N} \sigma_2(c^{*(k+1)})}. \tag{B.36}$$

Step 1D. Observe that

$$\text{(B.30)} \leqslant \frac{\|c^{*(k+1)} - \hat{c}^{(k+1)}\|_2^2}{\sigma_2^2((\hat{G}_{1:k} \otimes I_n)^T c^{*(k+1)})^2} \leqslant \frac{4 C_2^2 d^K n^{2K}}{N \sigma_2^2(c^{*(k+1)})} \leqslant \frac{\sqrt{C_3}}{2} \frac{\sqrt{n \log(dn) \|p^*\|_\infty}}{\sqrt{N} \sigma_2(c^{*(k+1)})}, \tag{B.37}$$

where second inequality follows from (B.11) and (B.27), and the last inequality hold for sufficiently large N in (3.39). Consequently, (B.29), (B.30), (B.36), and (B.37) together lead to

$$\|\tilde{G}_{k+1} \tilde{G}_{k+1}^T - \hat{G}_{k+1} \hat{G}_{k+1}\|_2 \leqslant \sqrt{C_3} \frac{\sqrt{n \log(dn) \|p^*\|_\infty}}{\sqrt{N} \sigma_2(c^{*(k+1)})}. \tag{B.38}$$

Step 1E. We complete the induction in this step. Note that

$$\text{(B.17)} \leqslant \|\hat{G}_{k+1} \hat{G}_{k+1}^T - \tilde{G}_{k+1} \tilde{G}_{k+1}^T\|_2^2 \|c^*\|_F^2 \leqslant C_3 \frac{n \log(dn) \|p^*\|_\infty}{N \sigma_2^2(c^{*(k+1)})} \|c^*\|_F^2,$$

where the first inequality follows from (B.22), and the last inequality follows from (B.38).

(B.17) and (B.18) together lead to

$$\|\hat{G}_{1:k+1}\hat{G}_{1:k+1}^T c^{*(k+1)} - c^{*(k+1)}\|_F^2 \leqslant C_3 \frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{N} \sum_{j=1}^{k+1} \frac{1}{\sigma_2^2(c^{*(j)})}.$$

This completes the induction. Note that by induction

$$\text{(B.8)} = \|\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)} - c^{*(d-1)}\|_F^2 \leqslant C_3 \frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{N} \sum_{j=1}^{d-1} \frac{1}{\sigma_2^2(c^{*(j)})}. \tag{B.39}$$

Step 2. We have

$$\begin{aligned}
\text{(B.9)} =& \|\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T(\hat{c}^{(d-1)} - c^{*(d-1)})\|_F^2 \\
\leqslant& 2\|(\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T - U_{d-1}^* U_{d-1}^{*T})(\hat{c}^{(d-1)} - c^{*(d-1)})\|_F^2 \qquad \text{(B.40)} \\
&+ 2\|U_{d-1}^* U_{d-1}^{*T}(\hat{c}^{(d-1)} - c^{*(d-1)})\|_F^2. \qquad \text{(B.41)}
\end{aligned}$$

Note that

$$\begin{aligned}
\text{(B.40)} \leqslant& 2\|\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T - U_{d-1}^* U_{d-1}^{*T}\|_2^2 \|\hat{c}^{(d-1)} - c^{*(d-1)}\|_F^2 \\
=& 2\|\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T - U_{d-1}^* U_{d-1}^{*T}\|_2^2 \|\hat{c} - c^*\|_F^2 \leqslant C_5 \frac{nB^{*3}\|p^*\|_\infty}{b^{*4}\sigma_2^2(c^{*(d-1)})} \frac{d\log(dn)}{N} \|\hat{c} - c^*\|_F^2 \\
\leqslant& C_5 \frac{nB^{*3}\|p^*\|_\infty}{b^{*4}\sigma_2^2(c^{*(d-1)})} \frac{d\log(dn)}{N} C_2^2 \left(\frac{d^K n^{2K}}{N}\right) \leqslant C_6 \frac{n\log(dn)\|p^*\|_\infty}{N}, \qquad \text{(B.42)}
\end{aligned}$$

where C_5, C_6 are universal constants. The second inequality follows from Lemma 32 given (B.39), the third inequality follows from (B.11), and the fourth inequality follows from (3.39).

In addition,

$$(\text{B.41}) = 2\|(U_{d-1}^{*}U_{d-1}^{*T} \otimes I_n)(\hat{c}^{(d)} - c^{*(d)})\|_F^2 \leqslant 2C_1^2 \frac{n\log(dn)\|p^*\|_\infty}{N}. \tag{B.43}$$

where the inequality follows from (B.10).

Therefore

$$(\text{B.9}) \leqslant C_7 \frac{n\log(dn)\|p^*\|_\infty}{N} \leqslant 2C_7 \frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{\sigma_2^2(c^{*(d-1)})N}, \tag{B.44}$$

for some universal constant C_7, where the first inequality follows from (B.42) and (B.43), and the second inequality follows from the fact that $\sigma_2(c^{*(d-1)}) \leqslant \|c^{*(d-1)}\|_F = \|c^*\|_F$. At this point, we combine (B.39) and (B.44) and obtain the final estimation for the relative error:

$$\frac{\|c^* - \hat{G}_{1:d}\|_F}{\|c^*\|_F} = \mathrm{O}_{\mathbb{P}}\left(\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N}\sum_{j=1}^{d-1}\frac{1}{\sigma_2^2(c^{*(j)})}}\right).$$

By further applying $\|p^*\|_\infty \leqslant 1 + B^*d$ from Assumption 4, we arrive at the results in Theorem 4. □

B.2.2 Proof of Corollary 1

Proof. Let $\bar{p}^* : \mathbb{R}^d \to \mathbb{R}$ be the orthogonal projection of p^*:

$$\bar{p}^*(x_1, \ldots, x_d) = \sum_{l_1=1}^{n} \cdots \sum_{l_d=1}^{n} c^*(l_1, \ldots, l_d)\phi_{l_1}(x_1)\cdots\phi_{l_d}(x_d).$$

Then

$$\begin{aligned}\|p^* - \tilde{p}\|^2_{L_2([0,1]^d)} &\leqslant \|p^* - \bar{p}^*\|^2_{L_2([0,1]^d)} + \|\bar{p}^* - \tilde{p}\|^2_{L_2([0,1]^d)} \\ &= \|p^* - \bar{p}^*\|^2_{L_2([0,1]^d)} + \|c^* - \hat{G}_{1:d}\|^2_{L_2([0,1]^d)} \\ &\leqslant C_8 B^{*2} d n^{-2\beta_0} + C_9 \frac{dnB^* \log(dn)}{N} \sum_{j=1}^{d-1} \frac{\|c^*\|_F^2}{\sigma_2^2(c^{*(j)})}\end{aligned}$$

where C_8 and C_9 are some universal constants. $\hat{G}_{1:d}$ is the output tensor train from TT-SVD (Algorithm 3). The last inequality follows from the error estimation for the approximation error $\|p^* - \bar{p}^*\|_{L_2}$ by Lemma 33 in the appendix and $\|c^* - \hat{G}_{1:d}\|_{L_2}$ by Theorem 4. Note that by Corollary 7, $\|c^*\|_F^2 = \mathrm{O}(B^* d)$. In addition, by (B.3), $\|p^*\|^2_{L_2([0,1]^d)} = 1 + \sum_{j=1}^d \|q_j^*\|^2_{L_2([0,1])}$. Therefore,

$$1 + b^{*2} d \leqslant \|p^*\|^2_{L_2([0,1]^d)} \leqslant 1 + B^{*2} d.$$

by Assumption 4. Consequently,

$$\begin{aligned}\frac{\|p^* - \tilde{p}\|^2_{L_2([0,1]^d)}}{\|p^*\|^2_{L_2([0,1]^d)}} &\leqslant C_{10} \left(\frac{B^*}{b^*}\right)^2 \left(n^{-2\beta_0} + \frac{dn \log(dn)}{N} \sum_{j=1}^{d-1} \frac{1}{\sigma_2^2(c^{*(j)})}\right). \\ &\leqslant C_{11} \left(\left(\frac{B^*}{b^*}\right)^2 n^{-2\beta_0} + \frac{B^{*4}}{b^{*6}} \frac{dn \log^2(dn)}{N} \right),\end{aligned}$$

where the last equality follows from Lemma 37 and the assumption that $B^* \geqslant 1$, and the third inequality follows from the fact that $\sum_{j=1}^d \frac{1}{j(d-j)} = \mathrm{O}(\frac{\log(d)}{d})$.

□

B.3 Useful Results

TT-SVD

Lemma 29. *Let $\hat{G}_{1:d}$ be the output tensor cores from Algorithm 3 with target rank r. Then*

$\hat{G}_{1:k} \in \mathbb{O}^{n^k \times r}$ *for* $k \in \{1, \cdots, d-1\}$,

Proof. We proceed by induction. Since $\hat{G}_1 = \mathrm{SVD}(\hat{c}^{(1)}, r) \in \mathbb{R}^{n \times r}$, it follows that $\hat{G}_1 \in \mathbb{O}^{n \times r}$. Assume $\hat{G}_{1:k-1} \in \mathbb{O}^{n^{k-1} \times r}$, by Lemma 47 we have

$$\hat{G}_{1:k} = \big(\hat{G}_{1:k-1} \otimes I_n\big)\hat{G}_k \in \mathbb{R}^{n^k \times r} \in \mathbb{O}^{n^k \times r}.$$

□

Lemma 30. *Let* $\hat{G}_{1:d}$ *be the output tensor cores from Algorithm 3 with target rank* r*. For* $k \in \{1, \cdots, d-1\}$*, it holds that*

$$\begin{aligned}
&\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 \\
=&\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - (\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)}\|_F^2 + \|\hat{G}_{1:k-1}\hat{G}_{1:k-1}^T c^{*(k-1)} - c^{*(k-1)}\|_F^2.
\end{aligned}$$

Proof. Observe that

$$\begin{aligned}
&\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 \\
=&\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - (\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)}\|_F^2 \\
+&\|(\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)} - c^{*(k)}\|_F^2 && \text{(B.45)} \\
+&\langle \hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)}, (\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)} - c^{*(k)} \rangle. && \text{(B.46)} \\
-&\langle (\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)}, (\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)} - c^{*(k)} \rangle. && \text{(B.47)}
\end{aligned}$$

Step 1. Note that

$$(\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)} = \{(\hat{G}_{1:k-1}\hat{G}_{1:k-1}^T) \otimes I_n\} c^{*(k)} = \hat{G}_{1:k-1}\hat{G}_{1:k-1}^T c^{*(k-1)}.$$

Therefore (B.45) $= \|\hat{G}_{1:k-1}\hat{G}_{1:k-1}^T c^{*(k-1)} - c^{*(k)}\|_F^2$.

Step 2. By Lemma 29, $\hat{G}_{1:k-1} \in \mathbb{O}^{n^{k-1}\times r}$. Then the kronecker product of two orthogonal matrices is still orthogonal, $\hat{G}_{1:k-1} \otimes I_n \in \mathbb{O}^{n^k \times rn}$. The details are in Lemma 46. It follows that

$$\begin{aligned}&(\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)} - c^{*(k)} = \big((\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T - I_{n^k}\big)c^{*(k)} \\ =&(\hat{G}_{1:k-1} \otimes I_n)_\perp \big[(\hat{G}_{1:k-1} \otimes I_n)_\perp\big]^T c^{*(k)}. \end{aligned} \tag{B.48}$$

Therefore

$$\text{(B.47)} = \langle (\hat{G}_{1:k-1} \otimes I_n)(\hat{G}_{1:k-1} \otimes I_n)^T c^{*(k)}, (\hat{G}_{1:k-1} \otimes I_n)_\perp \big[(\hat{G}_{1:k-1} \otimes I_n)_\perp\big]^T c^{*(k)} \rangle = 0.$$

Step 3. By (B.48), it follows that

$$\begin{aligned}\text{(B.46)} =&\langle (\hat{G}_{1:k-1} \otimes I_n)\hat{G}_k \hat{G}_{1:k}^T c^{*(k)}, (\hat{G}_{1:k-1} \otimes I_n)_\perp \big[(\hat{G}_{1:k-1} \otimes I_n)_\perp\big]^T c^{*(k)} \rangle \\ =&\langle \big[(\hat{G}_{1:k-1} \otimes I_n)_\perp\big]^T (\hat{G}_{1:k-1} \otimes I_n)\hat{G}_k \hat{G}_{1:k}^T c^{*(k)}, \big[(\hat{G}_{1:k-1} \otimes I_n)_\perp\big]^T c^{*(k)} \rangle = 0.\end{aligned}$$

This immediately leads to the desired result. □

Lemma 31. *Let $\hat{G}_{1:d}$ be the output tensor cores from Algorithm 3 with target rank r. Then*

$$\|c^* - \hat{G}_{1:d}\|_F^2 = \|c^{*(d-1)} - \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)}\|_F^2 + \|\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)} - \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T \hat{c}^{(d-1)}\|_F^2.$$

Proof. Since $\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T \hat{c}^{(d-1)}$ is a reshape matrix of $\hat{G}_{1:d}$, we have

$$\|c^* - \hat{G}_{1:d}\|_F^2 = \|c^{*(d-1)} - \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T \hat{c}^{(d-1)}\|_F^2.$$

It suffices to show that

$$\langle \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)} - c^{*(d-1)},\ \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T \hat{c}^{(d-1)} - \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)} \rangle = 0. \tag{B.49}$$

By Lemma 29, $\hat{G}_{1:d-1} \in \mathbb{O}^{n^{d-1} \times r}$. So

$$\hat{G}_{1:d-1}\hat{G}_{1:d-1}^T c^{*(d-1)} - c^{*(d-1)} = (\hat{G}_{1:d-1})_\perp (\hat{G}_{1:d-1}^T)_\perp c^{*(d-1)}.$$

Therefore $(\hat{G}_{1:d-1})_\perp^T \hat{G}_{1:d-1} = 0$ and so

$$\begin{aligned}(\text{B.49}) =& \langle ((\hat{G}_{1:d-1})_\perp (\hat{G}_{1:d-1})_\perp^T c^{*(d-1)},\ \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T(\hat{c}^{(d-1)} - c^{*(d-1)})\rangle \\ =& \langle ((\hat{G}_{1:d-1})_\perp^T c^{*(d-1)},\ (\hat{G}_{1:d-1})_\perp^T \hat{G}_{1:d-1}\hat{G}_{1:d-1}^T(\hat{c}^{(d-1)} - c^{*(d-1)})\rangle = 0.\end{aligned}$$

□

Lemma 32. *Suppose all the conditions in Theorem 4 holds. Let $\hat{G}_{1:d}$ be the output tensor cores from Algorithm 3 with target rank $r = 2$. Suppose that*

$$\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 \leqslant C\frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{N}\sum_{j=1}^{k}\frac{1}{\sigma_2^2(c^{*(j)})} \tag{B.50}$$

where C is a universal constant. Let U_k^* be defined as in Lemma 35. Note that by Lemma 35, $U_k^* U_k^{*T} \in \mathbb{R}^{n^k \times n^k}$ is the projection matrix onto the space $\mathrm{Col}(c^{*(k)})$. Then

$$\|\hat{G}_{1:k}\hat{G}_{1:k}^T - U_k^* U_k^{*T}\|_2 \leqslant C'\frac{\sqrt{nB^{*3}\|p^*\|_\infty}}{b^{*2}\sigma_2(c^{*(k)})}\sqrt{\frac{d\log(dn)}{N}}, \tag{B.51}$$

where C' a universal constant.

Proof. Note that by Lemma 37, for $k \in \{1, \cdots, d-1\}$, $\sigma_2^2(c^{*(k)}) \geqslant \frac{c_\sigma^2 k(d-k)}{d}$. In addition, by Corollary 7, $\|c^*\|_F^2 \leqslant C_c d$ where $C_c = B^* + 2$. Therefore by (B.50)

$$\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_F^2 \leqslant C\frac{n\log(dn)\|p^*\|_\infty \|c^*\|_F^2}{N}\frac{d}{c_\sigma^2}\sum_{j=1}^{k}\frac{1}{j(d-j)} \leqslant C''\frac{C_c n\|p^*\|_\infty}{c_\sigma^2}\frac{d\log(dn)}{N} \tag{B.52}$$

where we have used the fact that $\sum_{j=1}^{d}\frac{1}{j(d-j)} = \mathrm{O}(\frac{\log(d)}{d})$ and C'' is some universal constant.

By Lemma 45, $\mathrm{Rank}(\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)}) = 2$, and $\hat{G}_{1:k}\hat{G}_{1:k}^T$ is the projection matrix on $\mathrm{Col}(\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)})$. Consequently,

$$\begin{aligned}\|\hat{G}_{1:k}\hat{G}_{1:k}^T - U_k^* U_k^{*T}\|_2 \leqslant \frac{2\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_2}{\sigma_2(c^{*(k)}) - \|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_2} &\leqslant \frac{2\|\hat{G}_{1:k}\hat{G}_{1:k}^T c^{*(k)} - c^{*(k)}\|_2}{\frac{1}{2}\sigma_2(c^{*(k)})} \\ &\leqslant 4\frac{\sqrt{C'' C_c n\|p^*\|_\infty}}{c_\sigma \sigma_2(c^{*(k)})}\sqrt{\frac{d\log(dn)}{N}}\end{aligned} \tag{B.53}$$

where the first inequality follows from Theorem 21 and $\mathrm{Rank}(c^{*(k)}) = 2$, the second inequality follows from the assumption on sufficiently large sample size N as stated in (3.39), and the last inequality follows from (B.52). Furthermore, by the expressions of C_c in Corollary 7 and c_σ in Lemma 37, we have $\frac{\sqrt{C_c}}{c_\sigma} \sim \sqrt{\frac{B^{*3}}{b^{*4}}}$ can be bounded a universal constant, leading to the final estimation (B.51). □

Ground truth density function

Lemma 33. *Under Assumption 4, let*

$$\bar{p}^*(x_1, \cdots, x_d) = \sum_{l_1=1}^{n} \cdots \sum_{l_d=1}^{n} c^*(l_1, \cdots, l_d)\phi_{l_1}(x_1)\cdots\phi_{l_d}(x_d)$$

where $c^* \in \mathbb{R}^{n^d}$ *is the coefficient tensor formulated as* (B.1). *There exists an absolute constant* C *such that*

$$\|p^* - \bar{p}^*\|_{L_2([0,1]^d)} \leqslant Cn^{-\beta_0}\sum_{j=1}^{d}\|q_j^*\|_{H^{\beta_0}([0,1])}.$$

Proof. Direct calculations show that

$$\bar{p}^*(x_1, \cdots, x_d) = 1 + \sum_{l_1=1}^{n} c_1^*(l_1)\phi_{l_1}(x_1) + \cdots + \sum_{l_d=1}^{n} c_d^*(l_d)\phi_{l_d}(x_d).$$

where $c_j^* \in \mathbb{R}^n$ is defined in (B.4). Therefore

$$\|p^* - \bar{p}^*\|_{L_2([0,1]^d)} \leqslant \sum_{j=1}^{d} \left\| \sum_{l_j=1}^{n} c_j^*(l_j)\phi_{l_j} - q_j^* \right\|_{L_2([0,1])}.$$

Denote $\mathcal{S}_n$ the class of polynomial up degree n on $[0,1]$, and $\mathcal{P}_n$ the projection operator from $L_2([0,1])$ to $\mathcal{S}_n$ respect to L_2 norm. Since Legendre polynomials are orthogonal, (B.4) and (B.3)

$$\sum_{l_j=1}^{n} c_j^*(l_j)\phi_{l_j} = \mathcal{P}_n(q_j^*). \tag{B.54}$$

By the Legendre polynomial approximation theory shown in the appendix of [111], it follows that

$$\left\| \sum_{l_j=1}^{n} c_j^*(l_j)\phi_{l_j} - q_j^* \right\|_{L_2([0,1])} \leqslant C\|q_j^*\|_{H^{\beta_0}([0,1])} n^{-\beta_0} \tag{B.55}$$

for some absolute constant C which does not depend on c_j^* or q_j^*. The desired result follows immediately. □

Corollary 7. *Under Assumption 4, for sufficiently large n, there exists a constant C such that*

$$\|c^*\|_F^2 \leqslant C_c d, \quad \textit{where} \quad C_c = B^* + 2$$

where $c^* \in \mathbb{R}^{n^d}$ is the coefficient tensor formulated as (B.1).

Proof. Observe that

$$\begin{aligned}\|c^*\|_F^2 =& \|\mathbf{1}^d + c_1^* \otimes \mathbf{1}^{d-1} + \mathbf{1} \otimes c_2^* \otimes \mathbf{1}^{d-2} + \cdots + \mathbf{1}^{d-1} \otimes c_d^*\|_F^2 \\ =& \|\mathbf{1}^d\|_F^2 + \|c_1^* \otimes \mathbf{1}^{d-1}\|_F^2 + \cdots + \|\mathbf{1}^{d-1} \otimes c_d^*\|_F^2 = 1 + \sum_{j=1}^{d} \|c_j^*\|^2.\end{aligned}$$

where the first equality follows from (B.1) , and the second equality follows from orthogonality. In addition, note that

$$\|c_j^*\| = \big\| \sum_{l_j=1}^{n} c_j^*(l_j)\phi_{l_j} \big\|_{L_2([0,1])} \leqslant \|q_j^*\|_{L_2([0,1])} + C_1 \|q_j^*\|_{H^{\beta_0}([0,1])} n^{-\beta_0} \leqslant B^* + 1,$$

where the first inequality follows from (B.55), and the last inequality follows from Assumption 4 and n being sufficiently large. The desired result follows immediately. □

Row and column spaces of the coefficient tensor

Lemma 34. *Let $k \in \{1, \cdots, d-1\}$. Let $c^{*(k)} \in \mathbb{R}^{n^k} \times \mathbb{R}^{n^{d-k}}$ be the k-th unfolding of the tensor coefficient c^* as in* (B.1) *for $k = 1, \cdots, d-1$. The row and the column space of $c^{*(k)}$ are given by*

$$\mathrm{Col}(c^{*(k)}) = \mathrm{Span}\{\mathbf{1}^k,\ c_1^* \otimes \mathbf{1}^{k-1} + \mathbf{1} \otimes c_2^* \otimes \mathbf{1}^{k-2} + \cdots + \mathbf{1}^{k-1} \otimes c_k^*\}, \tag{B.56}$$

$$\mathrm{Row}(c^{*(k)}) = \mathrm{Span}\{\mathbf{1}^{d-k},\ c_{k+1}^* \otimes \mathbf{1}^{d-k-1} + \mathbf{1} \otimes c_{k+2}^* \otimes \mathbf{1}^{d-k-2} + \cdots + \mathbf{1}^{d-k-1} \otimes c_d^*\}. \tag{B.57}$$

Consequently, $\mathrm{Rank}(c^{*(k)}) \leqslant 2$. *Furthermore, if $q_j^* \neq 0$ for all $j \in \{1, \cdots, d\}$,* $\mathrm{Rank}(c^{*(k)}) = 2$.

Proof. It is direct to show that

$$\begin{aligned} c^{*(k)} =& (\mathbf{1}^k) \otimes \{\mathbf{1}^{d-k}\} + (c_1^* \otimes \mathbf{1}^{k-1}) \otimes \{\mathbf{1}^{d-k}\} + \cdots + (\mathbf{1}^{k-1} \otimes c_k^*) \otimes \{\mathbf{1}^{d-k}\} \\ &+ (\mathbf{1}^k) \otimes \{c_{k+1}^* \otimes \mathbf{1}^{d-k-1}\} + \cdots + (\mathbf{1}^k) \otimes \{\mathbf{1}^{d-k-1} \otimes c_d^*\}, \end{aligned} \tag{B.58}$$

where terms with ( ) are viewed as column vectors in $\mathbb{R}^{n^k}$, and terms with { } are viewed as

row vectors in $\mathbb{R}^{n^{d-k}}$. To show (B.56) holds for the column space, it suffices to rewrite

$$\begin{aligned} c^{*(k)} =&(\mathbf{1}^k) \otimes \{\mathbf{1}^{d-k} + c^*_{k+1} \otimes \mathbf{1}^{d-k-1} + \mathbf{1} \otimes c^*_{k+2} \otimes \mathbf{1}^{d-k-2} + \cdots + \mathbf{1}^{d-k-1} \otimes c^*_d\} \\ &+\big(c^*_1 \otimes \mathbf{1}^{k-1} + \mathbf{1} \otimes c^*_2 \otimes \mathbf{1}^{k-2} + \cdots + \mathbf{1}^{k-1} \otimes c^*_k\big) \otimes \{\mathbf{1}^{d-k}\}. \end{aligned}$$

Therefore (B.56) holds. The justification of (B.57) is similar and omitted for brevity. □

Lemma 35. *Let c^* be coefficient tensor as in* (B.1) *and c^*_j be as in* (B.4). *Denote $\boldsymbol{e}_1 = [1,0]^T$, $\boldsymbol{e}_2 = [0,1]^T$ and let the matrices $U^*_k \in \mathbb{R}^{n^k \times 2}$ and $V^*_k \in \mathbb{R}^{n^{d-k} \times 2}$ be such that*

$$U^*_k = \mathbf{1}^k \otimes \boldsymbol{e}_1 + \frac{c^*_1 \otimes \mathbf{1}^{k-1} + \mathbf{1} \otimes c^*_2 \otimes \mathbf{1}^{k-2} + \cdots + \mathbf{1}^{k-1} \otimes c^*_k}{\sqrt{\|c^*_1\|^2 + \cdots + \|c^*_k\|^2}} \otimes \boldsymbol{e}_2, \tag{B.59}$$

$$V^*_k = \mathbf{1}^{d-k} \otimes \boldsymbol{e}_1 + \frac{c^*_{k+1} \otimes \mathbf{1}^{d-k-1} + \mathbf{1} \otimes c^*_{k+2} \otimes \mathbf{1}^{d-k-2} + \cdots + \mathbf{1}^{d-k-1} \otimes c^*_d}{\sqrt{\|c^*_{k+1}\|^2 + \cdots + \|c^*_d\|^2}} \otimes \boldsymbol{e}_2. \tag{B.60}$$

Then $U^*_k U^{*T}_k \in \mathbb{R}^{n^k \times n^k}$ is the projection matrix onto the space $\mathrm{Col}(c^{*(k)})$, *and $V^*_k V^{*T}_k \in \mathbb{R}^{n^{d-k} \times n^{d-k}}$ is the projection matrix onto the space* $\mathrm{Row}(c^{*(k)})$.

Proof. Note that the set of vectors $\{\mathbf{1}^k, c^*_1 \otimes \mathbf{1}^{k-1}, \mathbf{1} \otimes c^*_2 \otimes \mathbf{1}^{k-2}, \mathbf{1}^{k-1} \otimes c^*_k\}$ are orthogonal. Direct calculations show that

$$U^{*T}_k U^*_k = \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 = I_2 \in \mathbb{R}^{2 \times 2}.$$

Therefore $U^*_k \in \mathbb{O}^{n^k \times 2}$. Since $\mathrm{Col}(U^*_k) = \mathrm{Col}(c^{*(k)})$, $U^*_k U^{*T}_k \in \mathbb{R}^{n^k \times n^k}$ is the projection matrix onto the space $\mathrm{Col}(c^{*(k)})$. The justification of $V^*_k V^{*T}_k$ is similar and omitted for brevity. □

For simplicity, in (B.59) and (B.60) we introduce the notations

$$u_k^* = \frac{c_1^* \otimes \mathbf{1}^{k-1} + \mathbf{1} \otimes c_2^* \otimes \mathbf{1}^{k-2} + \cdots + \mathbf{1}^{k-1} \otimes c_k^*}{\sqrt{\|c_1^*\|^2 + \cdots + \|c_k^*\|^2}} \in \mathbb{R}^{n^k}, \tag{B.61}$$

$$v_k^* = \frac{c_{k+1}^* \otimes \mathbf{1}^{d-k-1} + \mathbf{1} \otimes c_{k+2}^* \otimes \mathbf{1}^{d-k-2} + \cdots + \mathbf{1}^{d-k-1} \otimes c_d^*}{\sqrt{\|c_{k+1}^*\|^2 + \cdots + \|c_d^*\|^2}} \in \mathbb{R}^{n^{d-k}}. \tag{B.62}$$

Lemma 36. *Let u_k^* and v_k^* be as in* (B.61) *and* (B.62) *respectively. Denote*

$$\Phi_k^*(x_1, \cdots, x_k) = \frac{\sum_{l_1=1}^n c_1^*(l_1)\phi_{l_1}(x_1) + \cdots + \sum_{l_k=1}^n c_k^*(l_k)\phi_{l_k}(x_k)}{\sqrt{\|c_1^*\|^2 + \cdots + \|c_k^*\|^2}}, \tag{B.63}$$

$$\Psi_k^*(x_{k+1}, \cdots, x_d) = \frac{\sum_{l_{k+1}=1}^n c_{k+1}^*(l_{k+1})\phi_{l_{k+1}}(x_{k+1}) + \cdots + \sum_{l_d=1}^n c_d^*(l_d)\phi_{l_d}(x_d)}{\sqrt{\|c_{k+1}^*\|^2 + \cdots + \|c_d^*\|^2}}. \tag{B.64}$$

Then u_k^* is the coefficient tensor of Φ_k^*, and v_k^* is the coefficient tensor of Ψ_k^*. Then

$$\|\Phi_k^*\|_{L_2([0,1]^k)} = \|u_k^*\|_F = 1 \quad \textit{and} \quad \|\Psi_k^*\|_{L_2([0,1]^{d-k})} = \|v_k^*\|_F = 1.$$

In addition, suppose Assumption 4 holds. Then for sufficient large n such that

$$\|\Phi_k^*\|_\infty \leqslant C'\sqrt{k} \quad \textit{and} \quad \|\Psi_k^*\|_\infty \leqslant C'\sqrt{d-k} \quad \textit{where} \quad C' = \frac{4B^*}{b^*}.$$

Proof. To see that u_k^* is the coefficient tensor of $\Phi_k^*(x_1, \cdots, x_k)$, it suffices to observe that the $(l_1, \cdots, l_k)$-coordinate of u_k^* is $\langle \Phi_k^*, \phi_{l_1}^1 \cdots \phi_{l_k}^k \rangle$. Since the coefficient tensor is distance preserving, it holds that

$$\|\Phi_k^*\|_{L_2([0,1]^k)} = \|u_k^*\|_F = 1.$$

Next, we bound $\|\Phi_k^*\|_\infty$. By (B.54) and Theorem 23, it follows that there exists a constant

C_1 such that

$$\big\| \sum_{l_j=1}^{n} c_j^*(l_1)\phi_{l_j} \big\|_\infty = \|\mathcal{P}_n(q_j^*)\|_\infty \leqslant \|q_j^*\|_\infty + C_1 n^{-\beta_0}.$$

Therefore, the numerator of (B.63) is bounded by

$$\Big\| \sum_{l_1=1}^{n} c_1^*(l_1)\phi_{l_1} + \cdots + \sum_{l_k=1}^{n} c_k^*(l_k)\phi_{l_k} \Big\|_\infty \leqslant \sum_{j=1}^{k} \|q_j^*\|_\infty + C_1 k n^{-\beta_0} \leqslant 2kB^* \tag{B.65}$$

upon using Assumption 4 and sufficiently large n. Observe that

$$\|c_j^*\| = \big\| \sum_{l_j=1}^{n} c_j^*(l_j)\phi_{l_j} \big\|_{L_2([0,1])} = \|\mathcal{P}_n(q_j^*)\|_{L_2([0,1])} \geqslant \|q_j^*\|_{L_2([0,1])} - C_2 n^{-\beta_0} \geqslant b^* - C_2 n^{-\beta_0} \geqslant b^*/2$$

from some constant C_2, where the second equality follows from (B.54), the first inequality follows from Corollary 9, the second inequality follows from the Assumption 4 that $\min_{j\in\{1,\cdots,d\}} \|q_j^*\|_{L_2([0,1])} \geqslant b^*$, and the last inequality holds for sufficiently large n. Therefore, the denominator

$$\sqrt{\|c_1^*\|^2 + \cdots + \|c_k^*\|^2} \geqslant b^*\sqrt{k}/2 \tag{B.66}$$

Note that (B.65) and (B.66) immediately imply $\|\Phi_k^*\|_\infty \leqslant 4\frac{B^*}{b^*}\sqrt{k}$. The analysis for $\|\Psi_k^*\|_\infty$ is the same and therefore omitted. □

Singular Values of the reshaped coefficient tensor

Lemma 37. *Let c^* be defined as in* (B.1). *Denote*

$$a_k^* = \sum_{i=1}^{k} \|c_i^*\|^2 \quad \textit{and} \quad b_k^* = \sum_{i=k+1}^{d} \|c_i^*\|^2.$$

Then the k-th unfolding of coefficient tensor c^* has the smallest singular value formulated as

$$\sigma_2^2(c^{*(k)}) = \frac{1 + a_k^* + b_k^* - \sqrt{(1 + a_k^* + b_k^*)^2 - 4a_k^* b_k^*}}{2}$$

for $k = 1, \cdots, d$. Suppose in addition that Assumption 4 holds, for sufficient large n, $\sigma_2(c^{*(k)})$ is lower bounded by

$$\sigma_2(c^{*(k)}) \geqslant c_\sigma \sqrt{\frac{k(d-k)}{d}} \quad \textit{where} \quad c_\sigma = \frac{b^{*2}}{4\sqrt{1 + 4B^{*2}}}. \tag{B.67}$$

Proof. Let $k \in \{1, \cdots, d\}$. Denote

$$f_k^* = c_1^* \otimes \mathbf{1}^{k-1} + \mathbf{1} \otimes c_2^* \otimes \mathbf{1}^{k-2} + \cdots + \mathbf{1}^{k-1} \otimes c_k^*, \text{ and } g_k^* = c_{k+1}^* \otimes \mathbf{1}^{d-k-1} + \cdots + \mathbf{1}^{d-k} \otimes c_d^*.$$

Then one can easily check that $a_k^* = \|f^*\|_F^2$ and $b_k^* = \|g^*\|_F^2$ due to orthogonality, and we can write

$$c^* = (\mathbf{1}^k) \otimes \{\mathbf{1}^{d-k} + g_k^*\} + (f_k^*) \otimes \{\mathbf{1}^{d-k}\}.$$

where terms with ( ) are viewed as column vectors in $\mathbb{R}^{n^k}$, and terms with { } are viewed as row vectors in $\mathbb{R}^{n^{d-k}}$. Therefore

$$c^{*(k)} = \begin{bmatrix} \mathbf{1}^k & f_k^* \end{bmatrix} \begin{bmatrix} \mathbf{1}^{d-k} + g_k^* \\ \mathbf{1}^{d-k} \end{bmatrix}.$$

Observe that

$$c^{*(k)}(c^{*(k)})^T = \begin{bmatrix} \mathbf{1}^k & f_k^* \end{bmatrix} \begin{bmatrix} 1 + \|g_k^*\|_F^2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{1}^k \\ f_k^* \end{bmatrix} = \begin{bmatrix} \mathbf{1}^k & f_k^*/\|f_k^*\|_F \end{bmatrix} \begin{bmatrix} 1 + \|g_k^*\|_F^2 & \|f_k^*\|_F \\ \|f_k^*\|_F & \|f_k^*\|_F^2 \end{bmatrix} \begin{bmatrix} \mathbf{1}^k \\ f_k^*/\|f_k^*\|_F \end{bmatrix}$$

Note that $\begin{bmatrix}\mathbf{1}^k & f_k^*/\|f_k^*\|_F\end{bmatrix}$ is an orthogonal matrix in $\mathbb{O}^{n^k\times 2}$. Therefore the eigenvalues of $c^{*(k)}(c^{*(k)})^T$ coincide with the eigenvalues of $\begin{bmatrix}1+\|g_k^*\|_F^2 & \|f_k^*\|_F \\ \|f_k^*\|_F & \|f_k^*\|_F^2\end{bmatrix} \in \mathbb{R}^{2\times 2}$. Direct calculations show that

$$\sigma_2^2(c^{*(k)}) = \lambda_2(c^{*(k)}(c^{*(k)})^T) = \frac{1+a_k^*+b_k^* - \sqrt{(1+a_k^*+b_k^*)^2 - 4a_k^*b_k^*}}{2}.$$

For (B.67), note that by (B.55), there exists a constant C_1

$$\Big\| \sum_{l_j=1}^{n} c_j^*(l_j)\phi_{l_j} - q_j^* \Big\|_{L_2([0,1])} \leqslant C_1 n^{-\beta_0} \tag{B.68}$$

Since $\big\| \sum_{l_j=1}^n c_j^*(l_j)\phi_{l_j}\big\|_{L_2([0,1])} = \|c_j^*\|$, it follows that

$$\|q_j^*\|_{L_2([0,1])} - C_1 n^{-\beta_0} \leqslant \|c_j^*\| \leqslant \|q_j^*\|_{L_2([0,1])} + C_1 n^{-\beta_0}$$

It follows from Assumption 4 and (B.55) that

$$b^* \leqslant \|q_j^*\|_{L_2([0,1])} \leqslant B^*.$$

So for sufficiently large n, it follows that $\frac{b^*}{2} \leqslant \|c_j^*\| \leqslant 2B^*$. Consequently,

$$\frac{b^{*2}k}{4} \leqslant a_k^* \leqslant 4B^{*2}k \quad \text{and} \quad \frac{b^{*2}(d-k)}{4} \leqslant b_k^* \leqslant 4B^{*2}(d-k).$$

Therefore

$$\begin{aligned}\sigma_2^2(c^{*(k)}) =& \frac{1+a_k^*+b_k^*-\sqrt{(1+a_k^*+b_k^*)^2-4a_k^*b_k^*}}{2} \\ =& \frac{(1+a_k^*+b_k^*-\sqrt{(1+a_k^*+b_k^*)^2-4a_k^*b_k^*})(1+a_k^*+b_k^*+\sqrt{(1+a_k^*+b_k^*)^2-4a_k^*b_k^*})}{2(1+a_k^*+b_k^*+\sqrt{(1+a_k^*+b_k^*)^2-4a_k^*b_k^*})} \\ \geqslant& \frac{4a_k^*b_k^*}{4(1+a_k^*+b_k^*)} \geqslant \frac{\frac{1}{16}b^{*4}k(d-k)}{1+4B^{*2}d} \geqslant \frac{\frac{1}{16}b^{*4}k(d-k)}{(1+4B^{*2})d}\end{aligned}$$

□

Variance of cluster basis estimators

Lemma 38. *Suppose Assumption 4 holds. Let c^* be the coefficient tensor of p^*, and $\hat{c}$ be the s-cluster coefficient tensor estimator defined in Definition 4 with $s \geqslant 2$. Let U_k^* and V_k^* be defined as in Lemma 35. Then,*

$$\max_{k\in\{1,\cdots,d\}} \|(U_{k-1}^{*T} \otimes I_n)(c^{*(k)} - \hat{c}^{(k)}[s])V_k^*\|_F = \mathrm{O}_{\mathbb{P}}\left(\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N}}\right). \tag{B.69}$$

Proof. We suppose $s \geqslant 3$ in the proof, where the case of $s = 2$ is a simplified version from

the same calculations. Given any positive number t, we have

$$\begin{aligned}
&\mathbb{P}\left(\max_{k\in\{1,\cdots,d\}} \|(U_{k-1}^{*T}\otimes I_n)(c^{*(k)}-\hat{c}^{(k)}[s])V_k^*\|_F^2 \geqslant 4nt^2\right)\\
&=\mathbb{P}\left(\|(U_{k-1}^{*T}\otimes I_n)(c^{*(k)}-\hat{c}^{(k)}[s])V_k^*\|_F^2 \geqslant 4nt^2 \text{ for least one } k\in\{1,\cdots,d\}\right)\\
&\leqslant \sum_{k=1}^{d}\mathbb{P}\left(\|(U_{k-1}^{*T}\otimes I_n)(c^{*(k)}-\hat{c}^{(k)}[s])V_k^*\|_F^2 \geqslant 4nt^2\right) \qquad \text{(by union bound)}\\
&= \sum_{k=1}^{d}\mathbb{P}\left(\text{(B.73)} \geqslant 4nt^2\right)\\
&\leqslant \sum_{k=1}^{d}\sum_{l=1}^{n}\mathbb{P}\left(|\langle p^*-\hat{p},\Phi_{k-1}^*\phi_l^k\Psi_k^*\rangle| \geqslant t\right)+\mathbb{P}\left(|\langle p^*-\hat{p},\phi_l^k\rangle| \geqslant t\right)\\
&\qquad\qquad +\mathbb{P}\left(|\langle p^*-\hat{p},\Phi_{k-1}^*\phi_l^k\rangle| \geqslant t\right)+\mathbb{P}\left(|\langle p^*-\hat{p},\phi_l^k\Psi_k^*\rangle| \geqslant t\right).
\end{aligned} \tag{B.70}$$

We now bound the four terms in the double sums above by Lemma 43, which is based on Bernstein inequality.

We first bound

$$\mathbb{P}\left(|\langle p^*-\hat{p},\Phi_{k-1}^*\phi_l^k\Psi_k^*\rangle| \geqslant t\right).$$

Note that by Lemma 36, it holds that

$$\|\Phi_{k-1}^*\phi_l^k\Psi_k^*\|_{L_2([0,1]^d)}=\|\Phi_{k-1}^*\|_{L_2([0,1]^{k-1})}\|\phi_l\|_{L_2([0,1])}\|\Psi_{k+1}^*\|_{L_2([0,1]^{d-k})}=1,$$

and that

$$\|\Phi_{k-1}^*\phi_l^k\Psi_k^*\|_\infty=\|\Phi_{k-1}^*\|_\infty\|\phi_l\|_\infty\|\Psi_{k+1}^*\|_\infty\leqslant C'^2\sqrt{n(k-1)(d-k)}.$$

where $C' = \frac{4B^*}{b^*}$. Therefore, by Lemma 43,

$$\begin{aligned}\mathbb{P}\left(|\langle p^* - \hat{p}, \Phi_{k-1}^* \phi_l^k \Psi_k^* \rangle| \geqslant t\right) &\leqslant 2\exp\left(\frac{-Nt^2}{\|p^*\|_\infty \|\Phi_{k-1}^* \phi_l^k \Psi_k^*\|_{L_2([0,1]^d)}^2 + \|\Phi_{k-1}^* \phi_l^k \Psi_k^*\|_\infty t}\right) \\ &\leqslant 2\exp\left(\frac{-Nt^2}{\|p^*\|_\infty + C'^2\sqrt{n(k-1)(d-k)}t}\right) =: \frac{\epsilon}{4dn}.\end{aligned} \tag{B.71}$$

where we assume the right-hand side is $\frac{\epsilon}{4dn}$ for some sufficiently small constant ϵ. Solving the last line above yields the quadratic equation

$$Nt^2 - C'^2\sqrt{n(k-1)(d-k)}\log(\frac{8dn}{\epsilon})t - \log(\frac{8dn}{\epsilon})\|p^*\|_\infty = 0,$$

whose larger root can be bounded by

$$\begin{aligned}t_+ &= \frac{C'^2\sqrt{n(k-1)(d-k)}\log(\frac{8dn}{\epsilon}) + \sqrt{C'^4 n(k-1)(d-k)[\log(\frac{8dn}{\epsilon})]^2 + 4N\log(\frac{8dn}{\epsilon})\|p^*\|_\infty}}{2N} \\ &\leqslant C_\epsilon\sqrt{\frac{\log(dn)\|p^*\|_\infty}{N}} =: t_\epsilon\end{aligned}$$

upon choosing sufficiently large N. Here $C_\epsilon \sim \log(1/\epsilon)$ is some sufficiently large positive constant which only depends on ϵ. At this point, we have

$$\mathbb{P}\left(|\langle p^* - \hat{p}, \Phi_{k-1}^* \phi_l^k \Psi_k^* \rangle| \geqslant t_\epsilon\right) \leqslant \frac{\epsilon}{4dn}.$$

For the other three terms in the double sums, by Lemma 36 we have

$$\|\phi_l^k\|_\infty \leqslant \sqrt{n}, \quad \|\Phi_{k-1}^* \phi_l^k\|_\infty \leqslant \sqrt{n(k-1)}, \quad \|\phi_l^k \Psi_k^*\|_\infty \leqslant \sqrt{n(d-k)}.$$

Following the same calculations, we can show that with high probability

$$\begin{aligned}
&\mathbb{P}\left(\left|\langle p^* - \hat{p}, \phi_l^k \rangle\right| \geqslant t_\epsilon\right) \leqslant \frac{\epsilon}{4dn} \\
&\mathbb{P}\left(\left|\langle p^* - \hat{p}, \Phi_{k-1}^* \phi_l^k \rangle\right| \geqslant t_\epsilon\right) \leqslant \frac{\epsilon}{4dn}, \\
&\mathbb{P}\left(\left|\langle p^* - \hat{p}, \phi_l^k \Psi_k^* \rangle\right| \geqslant t_\epsilon\right) \leqslant \frac{\epsilon}{4dn}
\end{aligned}$$

with appropriate choice of N. Inserting the estimates above into (B.70), we conclude that

$$\mathbb{P}\left(\max_{k\in\{1,\cdots,d\}} \|(U_{k-1}^{*T} \otimes I_n)(c^{*(k)} - \hat{c}^{(k)}[s])V_k^*\|_F^2 \geqslant 4nt_\epsilon^2\right) \leqslant \epsilon$$

for any given positive ϵ. This indicates that, with high probability,

$$\max_{k\in\{1,\cdots,d\}} \|(U_{k-1}^{*T} \otimes I_n)(c^{*(k)} - \hat{c}^{(k)}[s])V_k^*\|_F \leqslant 2\sqrt{n}t_\epsilon = 2C_\epsilon\sqrt{\frac{n\log(dn)\|p^*\|_\infty}{N}}$$

□

Lemma 39. *Suppose Assumption 4 holds. Let c^* be the coefficient tensor of p^*, and $\hat{c}$ be the s-cluster coefficient tensor estimator defined in Definition 4 with $s \geqslant 2$. Then*

$$\|c^* - \hat{c}\|_F = \mathrm{O}_{\mathbb{P}}\left(\sqrt{\frac{d^s n^{2s}}{N}}\right). \tag{B.72}$$

Proof. It suffices to show that

$$\mathbb{E}\big(\|c^* - \hat{c}\|_F^2\big) = \mathrm{O}\left(\frac{d^s n^{2s}}{N}\right).$$

To this end, let $\mathcal{B}_{j,d}$ be the j-cluster basis functions defined in (3.5). Then by

$$\mathbb{E}(\|c^* - \hat{c}\|_F^2) = \sum_{j=0}^{s}\left\{\sum_{\varphi_0\in\mathcal{B}_{1,d}} \mathbb{E}(\langle \hat{p} - p^*, \varphi_0\rangle^2) + \cdots + \sum_{\varphi_s\in\mathcal{B}_{s,d}} \mathbb{E}(\langle \hat{p} - p^*, \varphi_s\rangle^2)\right\}$$

Let $g_s(x) = \prod_{k=1}^{s} \phi_{i_{j_k}}(x_{j_k})$ be a generic element in $\mathcal{B}_{s,d}$. Then since $\{\phi_l\}_{l=1}^{\infty}$ are Legendre polynomials, it follows that with $\|\phi_l\|_\infty \leqslant \sqrt{l}$ and therefore $\|g_s\|_\infty \leqslant n^{s/2}$. Consequently,

$$\begin{aligned}\mathbb{E}(\langle \hat{p} - p^*, g_s\rangle^2) = &\operatorname{Var}\left(\frac{1}{N}\sum_{i=1}^{N} g_s(x^{(i)})\right) = \frac{\operatorname{Var}(g_s(x^{(1)}))}{N} \\ \leqslant &\frac{\int g_s^2(x)p^*(x)\mathrm{d}x}{N} \leqslant \frac{\|g_s\|_\infty^2 \int p^*(x)\mathrm{d}x}{N} \leqslant \frac{n^s}{N},\end{aligned}$$

where the last equality follows from the fact that p^* is a density and so $\int p^*(x)dx = 1$. Since the cardinality of $\mathcal{B}_{s,d}$ is at most $n^s d^s$, it follows that $\sum_{\varphi_s \in \mathcal{B}_{s,d}} \mathbb{E}(\langle \hat{p} - p^*, \varphi_s\rangle^2) \leqslant \frac{d^s n^{2s}}{N}$. Since s is a bounded integers, it follows that

$$\mathbb{E}(\|c^* - \hat{c}\|_F^2) \leqslant \sum_{j=1}^{s} \frac{d^j n^{2j}}{N} = \mathrm{O}\left(\frac{d^s n^{2s}}{N}\right).$$

□

Lemma 40. *Let c^* be the coefficient tensor of p^*, and $\hat{c}[s]$ be the s-cluster coefficient tensor estimator defined in Definition 4 with $s \geqslant 2$. Let U_k^* and V_k^* be defined as in* (B.59) *and* (B.60) *respectively. Φ_k^* and Ψ_k^* are defined as in* (B.63) *and* (B.64) *respectively. The following equalities hold:*

- *If $s \geqslant 3$,*

$$\begin{aligned}&\|(U_{k-1}^{*T} \otimes I_n)(c^{*(k)} - \hat{c}[s]^{(k)})V_k^*\|_F^2 \\ = &\sum_{l=1}^{n}\left(\langle p^* - \hat{p}, \phi_l^k\rangle^2 + \langle p^* - \hat{p}, \Phi_{k-1}^*\phi_l^k\rangle^2 + \langle p^* - \hat{p}, \phi_l^k\Psi_k^*\rangle^2 + \langle p^* - \hat{p}, \Phi_{k-1}^*\phi_l^k\Psi_k^*\rangle^2\right);\end{aligned} \tag{B.73}$$

- *If $s = 2$,*

$$\|(U_{k-1}^{*T} \otimes I_n)(c^{*(k)} - \hat{c}[s]^{(k)})V_k^*\|_F^2 = \sum_{l=1}^{n} \left(\langle p^* - \hat{p}, \phi_l^k \rangle^2 + \langle p^* - \hat{p}, \Phi_{k-1}^* \phi_l^k \rangle^2 + \langle p^* - \hat{p}, \phi_l^k \Psi_k^* \rangle^2 \right). \tag{B.74}$$

Proof. For $l \in \{1, \cdots, n\}$, denote $\chi_l \in \mathbb{R}^n$ be the l-th canonical basis. That is, for $k \in \{1, \cdots, n\}$

$$\chi_l(k) = \begin{cases} 0, & \text{if } l \neq k; \\ 1, & \text{if } l = k. \end{cases}$$

Denote $\mathcal{T}$ the reshaped $\mathbb{R}^{2\times n\times 2}$ tensor of the matrix $(U_{k-1}^{*T} \otimes I_n)(c^{*(k)} - \hat{c}[s]^{(k)})V_k^*$. It follows from (B.59), (B.60), (B.61) and (B.62) that, for any $l \in \{1, \cdots, n\}$ the (a, l, b)-coordinate of $\mathcal{T}$ satisfies

$$\mathcal{T}_{a,l,b} = \begin{cases} \langle c^* - \hat{c}[s],\ \mathbf{1}^{k-1} \otimes \chi_l \otimes \mathbf{1}^{d-k} \rangle, & \text{if } a = 1, b = 1; \\ \langle c^* - \hat{c}[s],\ u_{k-1}^* \otimes \chi_l \otimes \mathbf{1}^{d-k} \rangle, & \text{if } a = 2, b = 1; \\ \langle c^* - \hat{c}[s],\ \mathbf{1}^{k-1} \otimes \chi_l \otimes v_k^* \rangle, & \text{if } a = 1, b = 2; \\ \langle c^* - \hat{c}[s],\ u_{k-1}^* \otimes \chi_l \otimes v_k^* \rangle, & \text{if } a = 2, b = 2. \end{cases}$$

By (B.63) and (B.64),

- $\mathbf{1}^{k-1} \otimes \chi_l \otimes \mathbf{1}^{d-k}$ is the coefficient tensor of $\phi_l(x_k) \in \text{Span}\{\mathcal{B}_{1,d}\}$;
- $u_{k-1}^* \otimes \chi_l \otimes \mathbf{1}^{d-k}$ is the coefficient tensor of $\Phi_{k-1}^*(x_1, \cdots, x_{k-1})\phi_l(x_k) \in \text{Span}\{\mathcal{B}_{2,d}\}$;
- $\mathbf{1}^{k-1} \otimes \chi_l \otimes v_k^*$ is the coefficient tensor of $\phi_l(x_k)\Psi_k^*(x_{k+1}, \cdots, x_d) \in \text{Span}\{\mathcal{B}_{2,d}\}$;
- $u_{k-1}^* \otimes \chi_l \otimes v_k^*$ is the coefficient tensor of $\Phi_{k-1}^*(x_1, \cdots, x_{k-1})\phi_l(x_k)\Psi_k^*(x_{k+1}, \cdots, x_d) \in \text{Span}\{\mathcal{B}_{3,d}\}$.

Step 1. Suppose $s \geqslant 3$. By Lemma 41,

- $\langle c^* - \hat{c}[s],\ \mathbf{1}^{k-1} \otimes \chi_l \otimes \mathbf{1}^{d-k} \rangle = \langle p^* - \hat{p}, \phi_l^k \rangle$;
- $\langle c^* - \hat{c}[s],\ u_{k-1}^* \otimes \chi_l \otimes \mathbf{1}^{d-k} \rangle = \langle p^* - \hat{p}, \Phi_{k-1}^* \phi_l^k \rangle$;
- $\langle c^* - \hat{c}[s],\ \mathbf{1}^{k-1} \otimes \chi_l \otimes v_k^* \rangle = \langle p^* - \hat{p}, \phi_l^k \Psi_k^* \rangle$;
- $\langle c^* - \hat{c}[s],\ u_{k-1}^* \otimes \chi_l \otimes v_k^* \rangle = \langle p^* - \hat{p}, \Phi_{k-1}^* \phi_l^k \Psi_k^* \rangle$.

This immediately leads to (B.73).

Step 2. Suppose $s = 2$. Note that $\Phi_{k-1}^* \phi_l^k \Psi_k^* \in \mathrm{Span}\{\mathcal{B}_{3,d}\}$. By Lemma 41, $\langle p^* - \hat{p}, \Phi_{k-1}^* \phi_l^k \Psi_k^* \rangle = 0$. The rest of the calculations are the same as **Step 1**. □

Lemma 41. *Suppose p^* satisfies* (3.38)*, and $\hat{c}[s]$ is the s-cluster coefficient tensor estimator of p^* in Definition 4, where $2 \leqslant s \leqslant d$. Suppose $t \in \{1, \cdots, d\}$ and $F : \mathbb{R}^d \to \mathbb{R}$ is any function such that $F \in \mathrm{Span}\{\mathcal{B}_{t,d}\}$. Let $f \in \mathbb{R}^{n^d}$ be the coefficient tensor of F as*

$$f(l_1, \cdots, l_d) := \langle F, \phi_{l_1}^1 \cdots \phi_{l_d}^d \rangle \quad \textit{for } 1 \leqslant l_1, \cdots, l_d \leqslant n.$$

Then

$$\langle c^* - \hat{c}[s], f \rangle = \begin{cases} 0, & \textit{if } t > s; \\ \langle p^* - \hat{p}, F \rangle, & \textit{if } t \leqslant s. \end{cases}$$

Proof. Observe that by orthogonality of $\{\mathcal{B}_{j,d}\}_{j=1}^{d}$

$$\begin{aligned}
\langle c^* - \hat{c}[s], f\rangle &= \sum_{l_1,\cdots,l_d=1}^{n} (c^* - \hat{c}[s])(l_1,\cdots,l_d)\cdot f(l_1,\cdots,l_d) \\
&= \sum_{l_1,\cdots,l_d=1}^{n} \langle p^* - \hat{p}_s,\ \phi_{l_1}^1 \cdots \phi_{l_d}^d\rangle\langle F,\ \phi_{l_1}^1 \cdots \phi_{l_d}^d\rangle \\
&= \sum_{\varphi_0\in\mathcal{B}_{0,d}} \langle p^* - \hat{p}_s, \varphi_0\rangle\langle F, \varphi_0\rangle + \cdots + \sum_{\varphi_d\in\mathcal{B}_{d,d}} \langle p^* - \hat{p}_s, \varphi_d\rangle\langle F, \varphi_d\rangle \\
&= \sum_{\varphi_0\in\mathcal{B}_{0,d}} \langle p^* - \hat{p}_s, \varphi_0\rangle\langle F, \varphi_0\rangle + \cdots + \sum_{\varphi_s\in\mathcal{B}_{s,d}} \langle p^* - \hat{p}_s, \varphi_s\rangle\langle F, \varphi_s\rangle. \qquad \text{(B.75)}
\end{aligned}$$

where in the second equality, $\hat{p}_s$ is defined in (B.5), the last equality follows the observation that $p^* \in \mathrm{Span}\{\mathcal{B}_{j,d}\}_{j=0}^{1}$ and $\hat{p}_s \in \mathrm{Span}\{\mathcal{B}_{j,d}\}_{j=0}^{s}$ with $s \geqslant 2$. Therefore,

- If $t > s$, then $\langle F, \varphi_s\rangle = 0$ when $\varphi_s \in \mathcal{B}_{s,d}$. So in this case (B.75) $= 0$;
- If $t \leqslant s$, then (B.75) $= \langle p^* - \hat{p}, F\rangle$ by Lemma 42.

□

Lemma 42. *Let Ω be a measurable set in arbitrary dimension. Let $\{\psi_k\}_{k=1}^{\infty}$ be a collection of orthonormal basis in $L_2(\Omega)$ and let $\mathcal{M}$ be a m-dimensional linear subspace of $L_2(\Omega)$ such that*

$$\mathcal{M} = \mathrm{Span}\{\psi_k\}_{k=1}^{m}.$$

Suppose $F, G \in \mathcal{M}$. Denote $\langle F, G\rangle = \int_\Omega F(x)G(x)dx$. Then

$$\langle F, G\rangle = \sum_{k=1}^{m} \langle F, \psi_k\rangle\langle G, \psi_k\rangle.$$

Proof. It suffices to observe that

$$\langle F, G\rangle = \sum_{k=1}^{\infty} \langle F, \psi_k\rangle\langle G, \psi_k\rangle = \sum_{k=1}^{m} \langle F, \psi_k\rangle\langle G, \psi_k\rangle,$$

where the first equality follows from the assumption that $\{\psi_k\}_{k=1}^{\infty}$ is a complete basis, and the second equality follows from the fact that $F \in \mathcal{M}$, and so $\langle F, \psi_k\rangle = 0$ if $k > m$. □

Lemma 43. *Suppose that $\{x^{(i)}\}_{i=1}^{N}$ are independently sampled from the density $p^* : \mathbb{R}^d \to \mathbb{R}^+$. Denote the empirical distribution $\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x^{(i)}}(x)$. Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be any non-random function. Then*

$$\mathbb{P}(\left|\langle \hat{p} - p^*, \varphi\rangle\right| \geqslant t) \leqslant 2\exp\left(\frac{-Nt^2}{\|p^*\|_\infty \|\varphi\|^2_{L_2([0,1]^d)} + \|\varphi\|_\infty t}\right).$$

Proof. It suffices to apply the Bernstein's inequality to the random variable $\langle \hat{p} - p^*, \varphi\rangle$. Observe that $\langle \hat{p} - p^*, \varphi\rangle = \frac{1}{N}\sum_{i=1}^{N}\left[\varphi(x^{(i)}) - \mathbb{E}[\varphi(x^{(i)})]\right]$. Since

$$\mathbb{E}(\left[\varphi(x^{(i)}) - \mathbb{E}[\varphi(x^{(i)})]\right]^2) = \mathrm{Var}[\varphi(x^{(i)})] \leqslant \mathbb{E}(\varphi^2(x^{(i)})) = \int \varphi^2(x)p^*(x)\mathrm{d}x \leqslant \|\varphi\|^2_{L_2([0,1]^d)}\|p^*\|_\infty$$

and that $\|\varphi(x^{(i)}) - \mathbb{E}[\varphi(x^{(i)})]\|_\infty \leqslant 2\|\varphi\|_\infty$, the desired result follows immediately from the Bernstein's inequality. □

Orthonormal matrices

Lemma 44. *Suppose $M \in \mathbb{R}^{n\times m}$ and $V \in \mathbb{O}^{n\times a}$ in the sense that $V^TV = I_a$. Suppose in addition that $V^TM = U\Sigma W^T$ is the SVD of V^TM, where $U \in \mathbb{O}^{a\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$ and $W \in \mathbb{O}^{r\times m}$. Then*

$$VV^TM = (VU)\Sigma W^T \tag{B.76}$$

is the SVD of VV^TM. *In particular,* $\mathrm{Rank}(V^TM) = \mathrm{Rank}(VV^TM)$ *and* $\sigma_j(V^TM) = \sigma_j(VV^TM)$ *for all* $1 \leqslant j \leqslant \mathrm{Rank}(V^TM)$.

Proof. Note that

$$(VU)^TVU = U^TV^TVU = U^TI_aU = U^TU = I_r.$$

Therefore $VU \in \mathbb{O}^{n\times r}$. Consequently, (B.76) holds by the uniqueness of SVD. Since the singular values both V^TM and VV^TM are both determined by Σ, the desired results follow.

□

Lemma 45. *Suppose* $M \in \mathbb{R}^{n\times m}$ *such that* $\mathrm{Rank}(M) = r$. *Suppose in addition that* $U \in \mathbb{O}^{n\times r}$ *and that* $\sigma_r(M) > 2\|UU^TM - M\|_F$. *Then*

- $\mathrm{Rank}(UU^TM) = \mathrm{Rank}(U^TM) = r$;
- $\mathrm{Row}(M) = \mathrm{Row}(UU^TM)$;
- $\sigma_r(U^TM) = \sigma_r(UU^TM) \geqslant \frac{\sigma_r(M)}{2}$.
- UU^T *is the projection matrix onto* $\mathrm{Col}(UU^TM)$. *In other words, the column vectors of* U *form a basis for* $\mathrm{Col}(UU^TM)$.

Proof. Note that $\mathrm{Row}(UU^TM) \subset \mathrm{Row}(M)$. In addition, by Theorem 20,

$$\sigma_r(UU^TM) \geqslant \sigma_r(M) - \|UU^TM - M\|_F > 0.$$

So $\mathrm{Rank}(UU^TM) \geqslant r$. Therefore $\dim(\mathrm{Row}(UU^TM)) \geqslant r$. Since $\mathrm{Row}(UU^TM) \subset \mathrm{Row}(M)$ and $\dim(\mathrm{Row}(M)) = \mathrm{Rank}(M) = r$, it follows that

$$\mathrm{Row}(UU^TM) = \mathrm{Row}(M) \quad \text{and} \quad \mathrm{Rank}(UU^TM) = r.$$

By Lemma 44, $\text{Rank}(U^T M) = \text{Rank}(UU^T M) = r$. In addition,

$$\sigma_r(U^T M) = \sigma_r(UU^T M) \geqslant \sigma_r(M) - \|UU^T M - M\|_F \geqslant \frac{\sigma_r(M)}{2},$$

where the equality follows from Lemma 44, and the first inequality follows from by Theorem 20.

Denote $U = [u_1, \cdots, u_r]$, where $\{u_k\}_{k=1}^r$ are orthogonal. Then

$$\text{Col}(UU^T M) \subset \text{Span}\{u_k\}_{k=1}^r.$$

Since $\dim(\text{Span}\{u_k\}_{k=1}^r) = \dim(\text{Col}(UU^T M)) = r$, it follows that $\text{Span}(\{u_k\}_{k=1}^r) = \text{Col}(UU^T M)$. Consequently $\{u_k\}_{k=1}^r$ form a basis for $\text{Col}(UU^T M)$.

□

Lemma 46. *Let $m, n, r, s \in \mathbb{Z}^+$ be positive integers. Suppose $U \in \mathbb{O}^{m\times r}$ and $W \in \mathbb{O}^{n\times s}$. Then $U \otimes W \in \mathbb{O}^{mn\times rs}$.*

Proof. It suffices to observe that

$$(U \otimes W)^T (U \otimes W) = (U^T \otimes W^T)(U \otimes W) = U^T U \otimes W^T W = I_r \otimes I_s = I_{rs}.$$

□

Lemma 47. *Let $m, r, s \in \mathbb{Z}^+$ be positive integers. Suppose $U \in \mathbb{O}^{m\times r}$ and $V \in \mathbb{O}^{rn\times s}$. Then $(U \otimes I_n)V \in \mathbb{O}^{mn\times s}$.*

Proof. It suffices to observe that

$$\begin{aligned}[(U \otimes I_n)V]^T(U \otimes I_n)V =& V^T(U^T \otimes I_n)(U \otimes I_n)V = V^T(U^TU \otimes I_n)V \\ =& V^T(I_r \times I_n)V = V^T I_{rn} V = I_s.\end{aligned}$$

□

Lemma 48. *Let $l, m, n, r \in \mathbb{Z}^+$ be positive integers. Let $U \in \mathbb{O}^{m\times r}$ and $U_\perp \in \mathbb{O}^{m\times(n-r)}$ be the orthogonal complement matrix of U. Then*

$$(U \otimes I_l)_\perp = U_\perp \otimes I_l.$$

Proof. It suffices to note that $U_\perp \otimes I_l \in \mathbb{O}^{ml\times(n-r)l}$ and that $\langle U \otimes I_l, U_\perp \otimes I_l\rangle = 0$. □

Matrix perturbation bound

Theorem 20. *Let A, $\hat{A}$ and E be three matrices in $\mathbb{R}^{n\times m}$ and that $\hat{A} = A + E$. Then*

$$|\sigma_r(A) - \sigma_r(\hat{A})| \leqslant \|E\|_2.$$

Proof. This is the well-known Weyl's inequality for singular values. □

Theorem 21. *Let A, $\hat{A}$ and E be three matrices in $\mathbb{R}^{n\times m}$ and that $\hat{A} = A + E$. Suppose the SVD of $A, \hat{A}$ satisfies*

$$A = \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} \hat{U} & \hat{U}_\perp \end{bmatrix} \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \hat{V}^T \\ \hat{V}_\perp^T \end{bmatrix}.$$

Here $r \leqslant \min\{n, m\}$, and $\Sigma_1, \hat{\Sigma}_1 \in \mathbb{R}^{r\times r}$ contain the leading r singular values of A and $\hat{A}$

respectively. If $\sigma_r(A) - \sigma_{r+1}(A) - \|E\|_2 > 0$*, then*

$$\|UU^T - \hat{U}\hat{U}^T\|_2 \leqslant \frac{2\|E\|_2}{\sigma_r(A) - \sigma_{r+1}(A) - \|E\|_2}.$$

Proof. This is the well-known Wedin's theorem. □

Lemma 49. *Let* A*,* $\hat{A}$ *and* E *be three matrices in* $\mathbb{R}^{n\times m}$ *and that* $\hat{A} = A + E$*. Suppose the SVD of* $A, \hat{A}$ *satisfies*

$$A = \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} \hat{U} & \hat{U}_\perp \end{bmatrix} \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \hat{V}^T \\ \hat{V}_\perp^T \end{bmatrix}.$$

Here $r \leqslant \min\{n, m\}$*, and* $\Sigma_1, \hat{\Sigma}_1 \in \mathbb{R}^{r\times r}$ *contain the leading* r *singular values of* A *and* $\hat{A}$ *respectively. Denote*

$$\alpha = \sigma_{\min}(U^T\hat{A}V),\ \beta = \|U_\perp^T\hat{A}V_\perp\|_2,\ z_{21} = \|U_\perp^T EV\|_2,\ \text{and } z_{12} = \|U^T EV_\perp\|_2,$$

If $\alpha^2 \geqslant \beta^2 + \min\{z_{21}^2, z_{12}^2\}$*, then*

$$\|UU^T - \hat{U}\hat{U}^T\|_2 \leqslant \frac{\alpha z_{21} + \beta z_{12}}{\alpha^2 - \beta^2 - \min\{z_{21}^2, z_{12}^2\}}.$$

Proof. See Theorem 1 of [24].

□

Corollary 8. *Let* A*,* $\hat{A}$ *and* E *be three matrices in* $\mathbb{R}^{n\times m}$ *and that* $\hat{A} = A + E$*. Suppose* $\mathrm{Rank}(A) = r$ *and* $\sigma_r(A) \geqslant 4\|E\|_2$*. Write the SVD of* $A, \hat{A}$ *as*

$$A = \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} \hat{U} & \hat{U}_\perp \end{bmatrix} \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \hat{V}^T \\ \hat{V}_\perp^T \end{bmatrix} \tag{B.77}$$

where $\Sigma_1, \hat{\Sigma}_1 \in \mathbb{R}^{r\times r}$ *contain the leading* r *singular values of* A *and* $\hat{A}$ *respectively. The for any* $\tilde{U} \in \mathbb{O}^{n\times r}$, $\tilde{V} \in \mathbb{O}^{m\times r}$ *such that*

$$\tilde{U}\tilde{U}^T = UU^T \quad \textit{and} \quad \tilde{V}\tilde{V}^T = VV^T,$$

there exists a universal constant C *such that*

$$\|\tilde{U}\tilde{U}^T - \hat{U}\hat{U}^T\|_2 \leqslant C\bigg(\frac{\|E\tilde{V}\|_2}{\sigma_r(A)} + \frac{\|E\|_2^2}{\sigma_r^2(A)}\bigg). \tag{B.78}$$

Proof. Let $\alpha, \alpha, z_{21}, z_{12}$ be defined as in Lemma 49. Observe that

$$\alpha = \sigma_{\min}(U^T\hat{A}V) = \sigma_r(U^T\hat{A}V) = \sigma_r(UU^T\hat{A}VV^T) = \sigma_r(UU^T(A+E)VV^T)$$

where the second equality follows from the fact that $U^T\hat{A}V \in \mathbb{R}^{r\times r}$, and the second equality follows from Lemma 44. So by Theorem 20,

$$|\alpha - \sigma_r(A)| \leqslant \|UU^TEVV^T\|_2 = \|E\tilde{V}\tilde{V}^T\|_2 = \|E\tilde{V}\|_2 \tag{B.79}$$

In addition, since $\mathrm{Rank}(A) = r$, it holds that $U_\perp^T A V_\perp = 0$. Therefore,

$$\beta = \|U_\perp^T(A+E)V_\perp\|_2 = \|U_\perp^T E V_\perp\|_2 \leqslant \|E\|_2.$$

In addition, we have that $z_{12} \leqslant \|E\|_2$ and that

$$z_{21} \leqslant \|EV\|_2 = \|EVV^T\|_2 = \|E\tilde{V}\tilde{V}^T\|_2 = \|E\tilde{V}\|_2,$$

where the first and the last equality both follow from Lemma 44. Consequently

$$\begin{aligned}\alpha^2 - \beta^2 - \min\{z_{21}^2, z_{12}^2\} \geqslant & \frac{1}{2}\sigma_r^2(A) - 2\|E\tilde{V}\|_2^2 - \|E\|_2^2 - \|E\|_2^2 \\ \geqslant & \frac{1}{2}\sigma_r^2(A) - 4\|E\|_2^2 \geqslant \frac{1}{4}\sigma_r^2(A),\end{aligned}$$

where the first inequality follows from (B.79), the second inequality follows from $\|E\tilde{V}\|_2 \leqslant \|E\|_2$, and the last inequality follows from $\sigma_r(A) \geqslant 4\|E\|_2$. Therefore by Lemma 49,

$$\begin{aligned}&\|\tilde{U}\tilde{U}^T - \hat{U}\hat{U}^T\|_2 = \|UU^T - \hat{U}\hat{U}^T\|_2 \leqslant \frac{\alpha z_{21} + \beta z_{12}}{\alpha^2 - \beta^2 - \min\{z_{21}^2, z_{12}^2\}} \\ \leqslant & \frac{(\sigma_r(A) + \|E\tilde{V}\|_2)\|E\tilde{V}\|_2 + \|E\|_2^2}{\frac{1}{4}\sigma_r^2(A)} \leqslant \frac{4\|E\tilde{V}\|_2}{\sigma_r(A)} + \frac{8\|E\|_2^2}{\sigma_r^2(A)},\end{aligned}$$

where $\|E\tilde{V}\|_2 \leqslant \|E\|_2$ is used in the last inequality. □

Polynomial Approximation Theory

Let $C^{\beta_0}([0,1])$ denote the class of functions with continuous derivatives up to order $\beta_0 \geqslant 1$.

Theorem 22. *For any $f \in C^{\beta_0}([0,1])$, there exists a polynomial p_f of degree at most n such that*

$$\|f - p_f\|_\infty \leqslant C\|f^{(\beta_0)}\|_\infty n^{-\beta_0}, \tag{B.80}$$

where C is an absolute constant which does not depend on f.

Proof. This is the well-known Jackson's Theorem for polynomial approximation. See Section 7.2 of [50] for a proof. □

Corollary 9. *Let $f \in C^{\beta_0}([0,1])$. Denote $\mathcal{S}_n$ the class of polynomial up degree n on $[0,1]$, and $\mathcal{P}_n$ the projection operator from $L_2([0,1])$ to $\mathcal{S}_n$ respect to L_2 norm. Then there exists*

a constant C which does not depend on f such that

$$\|f - \mathcal{P}_n(f)\|_{L_2([0,1])} \leqslant C\|f^{(\beta_0)}\|_\infty n^{-\beta_0}.$$

Proof. Let p_f be the polynomial satisfying (B.80). Then

$$\|f - \mathcal{P}_n(f)\|_{L_2([0,1])} \leqslant \|f - p_f\|_{L_2([0,1])} \leqslant \|f - p_f\|_\infty \leqslant C\|f^{(\beta_0)}\|_\infty n^{-\beta_0}.$$

□

Theorem 23. *Let $f \in C^{\beta_0}([0,1])$. Denote $\mathcal{S}_n$ the class of polynomial up degree n on $[0,1]$, and $\mathcal{P}_n$ the projection operator from $L_2([0,1])$ to $\mathcal{S}_n$ respect to L_2 norm. Then there exists an constant C only depending on f such that*

$$\|f - \mathcal{P}_n(f)\|_\infty \leqslant Cn^{-\beta_0}.$$

Proof. See Theorem 4.8 of [225]. □

APPENDIX C

APPENDIX OF HIERARCHICAL TENSOR SKETCHING

C.1 Preliminaries

We first provide a few useful lemmas and corollaries for conducting the analysis.

Lemma 50. *(Theorem 4.1 in [230]) Suppose two matrices A, B have the same rank $r(A) = r(B)$, then*

$$\|A^\dagger - B^\dagger\|_2 \leq 3\|A^\dagger\|_2\|B^\dagger\|_2\|A - B\|_2$$

Corollary 10. *(Corollary to Matrix Bernstein inequality, cf. Corollary 6.2.1 in [213]) Let $A^* \in \mathbb{R}^{m_1 \times m_2}$ be a matrix, and let $\{A^{\{j\}} \in \mathbb{R}^{m_1 \times m_2}\}_{j=1}^N$ be a sequence of i.i.d. matrices with $\mathbb{E}[A^{\{j\}}] = A^*$. Denote $\hat{A} = \frac{1}{N}\sum_{j=1}^N A^{\{j\}}$. Suppose there exists a constant M such that $\|A^{\{j\}}\|_2 \leq M$ for every j, then the following probability inequality holds for any $t > 0$:*

$$\mathbb{P}[\|\hat{A} - A^*\|_2 \geq t] \leq (m_1 + m_2)\exp(\frac{-Nt^2/2}{M^2 + 2Mt/3})$$

Corollary 11. *Assume $D \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is a three-dimensional tensor and $A \in \mathbb{R}^{r_1 \times n_1}, B \in \mathbb{R}^{r_2 \times n_2}, C \in \mathbb{R}^{n_1 \times n_2 \times r_3}$ satisfies $D(:,:,\gamma) = AC(:,:,\gamma)B^T$ for $\gamma \in [r_3]$. Then the following upper bound holds*

$$\|D\|_F \leq \|A\|_2\|B\|_2\|C\|_F$$

Proof. For each γ, $\|D(:,:,\gamma)\|_F = \|AC(:,:,\gamma)B^T\|_F \leq \|A\|_2\|C(:,:,\gamma)B^T\|_F = \|A\|_2\|BC(:,:,\gamma)^T\|_F \leq \|A\|_2\|B\|_2\|C(:,:,\gamma)^T\|_F = \|A\|_2\|B\|_2\|C(:,:,\gamma)\|_F$. Then the following inequality holds,

$$\|D\|_F^2 = \sum_\gamma \|D(:,:,\gamma)\|_F^2 \leq \sum_\gamma \|A\|_2^2\|B\|_2^2\|C(:,:,\gamma)\|_F^2 = \|A\|_2^2\|B\|_2^2\|C\|_F^2.$$

Therefore, $\|D\|_F \leq \|A\|_2\|B\|_2\|C\|_F$. □

C.2 Proof of Theorem 7

We first summarize the organization of this subsection. First, we study the perturbation error of tensor core $\|\hat{G}_k^{(l)} - G_k^{(l)*}\|_F$ in each level. Then based on the hierarchical structure of the tensor-network, we could recursively bound the error $\|\tilde{p}_k^{(l-1)} - p_k^{(l-1)*}\|_F$ based on lower level error $\|\tilde{p}_k^{(l)} - p_k^{(l)*}\|_F$ in the hierarchical tensor-network structure, and thus bound the final error $\|\tilde{p} - p^*\|_F$ in the end. We split the whole proof into three lemmas.

Lemma 51. $\|\hat{G}_k^{(l-1)} - G_k^{(l-1)*}\|_F \leq \kappa_G \epsilon_A$ *for* $k \in [2^l], l \in [L]$ *where* $\kappa_G = c_{\hat{A}^\dagger}^2 + 6c_{A^\dagger} c_{\hat{A}^\dagger}^2 c_p + 6c_{A^\dagger}^2 c_{\hat{A}^\dagger} c_p$.

Proof. Based on (4.11), $G_k^{(l-1)*}$ for $l = 1, \cdots, L$ takes the form

$$G_k^{(l-1)*} = \left(\mathcal{P}_{r^{(l)}}(A_{2k-1}^{(l)*})\right)^\dagger B_k^{(l-1)*} \left(\mathcal{P}_{r^{(l)}}(A_{2k}^{(l)*})^T\right)^\dagger = \left(A_{2k-1}^{(l)*}\right)^\dagger B_k^{(l-1)*} \left(A_{2k}^{(l)*T}\right)^\dagger, k \in [2^{l-1}]. \tag{C.1}$$

The second equality is due to Proposition 1. Similarly, we have a corresponding representation $\hat{G}_k^{(l-1)}$ for empirical distribution $\hat{p}$:

$$\hat{G}_k^{(l-1)} = \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger \hat{B}_k^{(l-1)} \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k}^{(l)})^T\right)^\dagger, k \in [2^{l-1}]. \tag{C.2}$$

To analyze the error of $\hat{G}_k^{(l-1)}$, we need to analyze the error of $\left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger$, $\left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k}^{(l)})\right)^\dagger$ and $\hat{B}_k^{(l-1)}$, respectively. Here we use Lemma 50 for $\left(\mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\right)^\dagger$, $(i = 2k-1, 2k)$

$$\| \left(A_i^{(l)*}\right)^\dagger - \left(\mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\right)^\dagger \|_2 \leq 3 \| \left(A_i^{(l)*}\right)^\dagger \|_2 \| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\right)^\dagger \|_2 \| A_i^{(l)*} - \mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)}) \|_2, \tag{C.3}$$

where

$$\begin{aligned}\|A_i^{(l)*} - \mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\|_2 &= \|A_i^{(l)*} - \hat{A}_i^{(l)} + \hat{A}_i^{(l)} - \mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\|_2 \\ &\leq \|A_i^{(l)*} - \hat{A}_i^{(l)}\|_2 + \|\hat{A}_i^{(l)} - \mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\|_2 \leq 2\|A_i^{(l)*} - \hat{A}_i^{(l)}\|_2 \leq 2\epsilon_A.\end{aligned} \tag{C.4}$$

The second inequality follows the fact that $\mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})$ is the best rank-$r^{(l)}$ approximation of $\hat{A}_i^{(l)}$ in terms of spectral norm and $\text{rank}(A_i^{(l)*}) = r^{(l)}$. Plugging this into (C.3), we get

$$\| \left(A_i^{(l)*}\right)^\dagger - \left(\mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\right)^\dagger \|_2 \leq 6c_{A^\dagger}c_{\hat{A}^\dagger}\epsilon_A, \tag{C.5}$$

where we use the definition in (4.21) for bounding $\| \left(A_i^{(l)*}\right)^\dagger \|_2$ and $\| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_i^{(l)})\right)^\dagger \|_2$.

Next, we upper bound the error of $\hat{B}_k^{(l-1)}$ (defined via (4.8), as mentioned in the introduction of this section):

$$\begin{aligned}\|\hat{B}_k^{(l-1)} - B_k^{(l-1)*}\|_F &= \|S_{2k-1}^{(l)T}\hat{p}_k^{(l-1)}S_{2k}^{(l)} - S_{2k-1}^{(l)T}p_k^{(l-1)*}S_{2k}^{(l)}\|_F \leq c_S^2\|\hat{p}_k^{(l-1)} - p_k^{(l-1)*}\|_F \\ &\leq c_S^2\| \left(S_k^{(l-1)T}\right)^\dagger \|_2\|\hat{A}_k^{(l-1)} - A_k^{(l-1)*}\|_F \leq c_S^2 c_{S^\dagger}\epsilon_A = \epsilon_A,\end{aligned} \tag{C.6}$$

where the first inequality is due to Corollary 11 and the second inequality is from (4.7), the definition of $\hat{A}_k^{(l-1)}$ and $A_k^{(l-1)*}$. Here $c_S = c_{S^\dagger} = 1$ since $S_{2k-1}^{(l)}, S_{2k}^{(l)}, S_k^{(l-1)}$ are orthogonal matrices in the assumption of the theorem. Besides,

$$\|B_k^{(l-1)*}\|_F = \|S_{2k-1}^{(l)T}p_k^{(l-1)*}S_{2k}^{(l)}\|_F \leq c_S^2\|p_k^{(l-1)*}\|_F \leq c_p, \tag{C.7}$$

where the first inequality is from Corollary 11. Therefore, we could have the following bound:

$$\begin{aligned}
\|\hat{G}_k^{(l-1)} - G_k^{(l-1)*}\|_F &= \| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger \hat{B}_k^{(l-1)} \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k}^{(l)})^T\right)^\dagger - \left(A_{2k-1}^{(l)*}\right)^\dagger B_k^{(l-1)*} \left(A_{2k}^{(l)*T}\right)^\dagger \|_F \\
&\leq \| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger \hat{B}_k^{(l-1)} \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k}^{(l)})^T\right)^\dagger - \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger B_k^{(l-1)*} \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k}^{(l)})^T\right)^\dagger \|_F \\
&+ \| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger B_k^{(l-1)*} \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k}^{(l)})^T\right)^\dagger - \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger B_k^{(l-1)*} \left(A_{2k}^{(l)*T}\right)^\dagger \|_F \\
&+ \| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger B_k^{(l-1)*} \left(A_{2k}^{(l)*T}\right)^\dagger - \left(A_{2k-1}^{(l)*}\right)^\dagger B_k^{(l-1)*} \left(A_{2k}^{(l)*T}\right)^\dagger \|_F \\
&\leq c_{\hat{A}^\dagger}^2 \|\hat{B}_k^{(l-1)} - B_k^{(l-1)*}\|_F + c_{\hat{A}^\dagger} c_p \| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k}^{(l)})^T\right)^\dagger - \left(A_{2k}^{(l)*T}\right)^\dagger \|_2 \\
&+ c_{A^\dagger} c_p \| \left(\mathcal{P}_{r^{(l)}}(\hat{A}_{2k-1}^{(l)})\right)^\dagger - \left(A_{2k-1}^{(l)*}\right)^\dagger \|_2 \\
&\leq (c_{\hat{A}^\dagger}^2 + 6c_{A^\dagger} c_{\hat{A}^\dagger}^2 c_p + 6c_{A^\dagger}^2 c_{\hat{A}^\dagger} c_p)\epsilon_A, \qquad \text{(C.8)}
\end{aligned}$$

where the first equality is from (C.1) and (C.2), the second inequality is based on Corollary 11, and the third inequality is from (C.5). Besides,

$$\|G_k^{(l-1)*}\|_F = \|(A_{2k-1}^{(l)*})^\dagger B_k^{(l-1)*} (A_{2k}^{(l)*T})^\dagger\|_F \leq c_{A^\dagger}^2 c_p, \qquad \text{(C.9)}$$

following Corollary 11 and (C.7). □

The next step is to analyze the error of the hierarchical tensor-network in estimating p^*.

Lemma 52. *If* $c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 > 1$, *then* $\|\tilde{p} - p^*\|_F \leq (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^L \left(1 + \frac{c_{\tilde{p}}^2 \kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1}\right) \epsilon_A$ *where* κ_G *is defined in Lemma 51.*

Proof. We want to prove that error $\|\tilde{p}_k^{(l)} - p_k^{(l)*}\|_F$ has the following form:

$$\|\tilde{p}_k^{(l)} - p_k^{(l)*}\|_F \leq \kappa_{p^{(l)}} \epsilon_A, \qquad \text{(C.10)}$$

for $l = 0, \cdots, L$ with some $\kappa_{p^{(l)}}$. Applying this to $\tilde{p}_1^{(0)} = \tilde{p}, p_1^{(0)*} = p^*$ we can obtain the desired conclusion. We obtain the form of $\kappa_{p^{(l)}}$ in (C.10) by induction.

At L-th level, we have $\tilde{p}_k^{(L)}(\mathcal{I}_k^{(L)},:) = \hat{p}(\mathcal{I}_k^{(L)},:)T_k^{(L)}$ for $k \in [2^L]$ from (4.20). Therefore,

$$\|\tilde{p}_k^{(L)} - p_k^{(L)*}\|_F = \|\hat{p}(\mathcal{I}_k^{(L)},:)T_k^{(L)} - p^*(\mathcal{I}_k^{(L)},:)T_k^{(L)}\|_F \leq \| \left(S_k^{(L)}\right)^\dagger \|_2 \|\hat{A}_k^{(L)} - A_k^{(L)*}\|_F \leq c_{S^\dagger}\epsilon_A = \epsilon_A, \tag{C.11}$$

where the first inequality is due to (4.7), and the definitions of $\hat{A}_k^{(l)}$ and $A_k^{(l)*}$ are given in the introduction of this section. Thus, $\|\tilde{p}_k^{(L)} - p_k^{(L)*}\|_F \leq \kappa_{p^{(L)}}\epsilon_A$ holds for any k where $\kappa_{p^{(L)}} = 1$.

Next, we derive the error (C.10) for $(l-1)$-th level, in terms of the the error of l-th level. Assuming $\|\tilde{p}_k^{(l)} - p_k^{(l)*}\|_F \leq \kappa_{p^{(l)}}\epsilon_A$ holds for $k \in [2^l]$. Based on (4.20), we have

$$\tilde{p}_k^{(l-1)} = \tilde{p}_{2k-1}^{(l)}\hat{G}_k^{(l-1)}\tilde{p}_{2k}^{(l)T}, p_k^{(l-1)*} = p_{2k-1}^{(l)*}G_k^{(l-1)*}p_{2k}^{(l)*T}, k \in [2^{l-1}], \tag{C.12}$$

and

$$\begin{aligned}
\|\tilde{p}_k^{(l-1)} - p_k^{(l-1)*}\|_F &= \|\tilde{p}_{2k-1}^{(l)}\hat{G}_k^{(l-1)}\tilde{p}_{2k}^{(l)T} - p_{2k-1}^{(l)*}G_k^{(l-1)*}p_{2k}^{(l)*T}\|_F \\
&\leq \|\tilde{p}_{2k-1}^{(l)}\hat{G}_k^{(l-1)}\tilde{p}_{2k}^{(l)T} - \tilde{p}_{2k-1}^{(l)}G_k^{(l-1)*}\tilde{p}_{2k}^{(l)T}\|_F \\
&+ \|\tilde{p}_{2k-1}^{(l)}G_k^{(l-1)*}\tilde{p}_{2k}^{(l)T} - p_{2k-1}^{(l)*}G_k^{(l-1)*}\tilde{p}_{2k}^{(l)T}\|_F \\
&+ \|p_{2k-1}^{(l)*}G_k^{(l-1)*}\tilde{p}_{2k}^{(l)T} - p_{2k-1}^{(l)*}G_k^{(l-1)*}p_{2k}^{(l)*T}\|_F \\
&\leq c_{\tilde{p}}^2\|\hat{G}_k^{(l-1)} - G_k^{(l-1)*}\|_F + c_{A^\dagger}^2 c_p c_{\tilde{p}}\|\tilde{p}_{2k-1}^{(l)} - p_{2k-1}^{(l)*}\|_F + c_{A^\dagger}^2 c_p^2\|\tilde{p}_{2k}^{(l)} - p_{2k}^{(l)*}\|_F \\
&\leq (c_{\tilde{p}}^2\kappa_G + c_{A^\dagger}^2 c_p c_{\tilde{p}}\kappa_{p^{(l)}} + c_{A^\dagger}^2 c_p^2\kappa_{p^{(l)}})\epsilon_A, k \in [2^{l-1}],
\end{aligned} \tag{C.13}$$

where the second inequality is from Corollary 11 and the third inequality is from Lemma 51. Therefore, $\|\tilde{p}_k^{(l-1)} - p_k^{(l-1)*}\|_F \leq \kappa_{p^{(l-1)}}\epsilon_A$ holds for $\kappa_{p^{(l-1)}} = c_{\tilde{p}}^2\kappa_G + (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)\kappa_{p^{(l)}}$. By induction, this equality holds for every l and we have

$$\kappa_{p^{(l-1)}} + \frac{c_{\tilde{p}}^2\kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1} = (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)\left(\kappa_{p^{(l)}} + \frac{c_{\tilde{p}}^2\kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1}\right). \tag{C.14}$$

Furthermore,

$$\kappa_{p^{(0)}} + \frac{c_{\tilde{p}}^2 \kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1} = (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^L \left(\kappa_{p^{(L)}} + \frac{c_{\tilde{p}}^2 \kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1} \right), \quad \text{(C.15)}$$

with the assumption $c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 > 1$, we have

$$\begin{aligned} \kappa_{p^{(0)}} &\leq (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^L \left(\kappa_{p^{(L)}} + \frac{c_{\tilde{p}}^2 \kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1} \right) \\ &= (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^L \left(1 + \frac{c_{\tilde{p}}^2 \kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1} \right). \end{aligned} \quad \text{(C.16)}$$

In the end, we have

$$\|\tilde{p} - p^*\|_F \leq \kappa_{p^{(0)}} \epsilon_A = (c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2)^{\log_2 d} \left(1 + \frac{c_{\tilde{p}}^2 \kappa_G}{c_{A^\dagger}^2 c_p c_{\tilde{p}} + c_{A^\dagger}^2 c_p^2 - 1} \right) \epsilon_A \quad \text{(C.17)}$$

with $L = \log_2 d$ mentioned in the introduction of this section. □

In the following lemma, we state what the upper-bound ϵ_A is, which completes the proof for Theorem 7.

Lemma 53. *For $0 < \delta < 1$, the following inequality for ϵ_A holds with probability at least $1 - \delta$:*

$$\epsilon_A \leq \frac{3\sqrt{\tilde{r}_{max}} \log(\frac{2\tilde{r}_{max} d \log_2 d}{\delta})}{\sqrt{N}} \quad \text{(C.18)}$$

Proof. We first consider $\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F$ for each k, l. This can be bounded via spectral norm: $\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F \leq \sqrt{\tilde{r}^{(l)}} \|\hat{A}_k^{(l)} - A_k^{(l)*}\|_2$. The error in spectral norm can be further bounded via matrix Bernstein inequality in Corollary 10.

More specifically, by definition $\hat{A}_k^{(l)}$ takes the form

$$\hat{A}_k^{(l)} = S_k^{(l)T}(\mathcal{I}_k^{(l)}, :) \hat{p}(\mathcal{I}_k^{(l)}; \mathcal{J}_k^{(l)}) T_k^{(l)}(\mathcal{J}_k^{(l)}, :) = \frac{1}{N} \sum_{j=1}^{N} S_k^{(l)}(\boldsymbol{y}^j_{C_k^{(l)}}, :)^T T_k^{(l)}(\boldsymbol{y}^j_{C_k^{(l)}}, :), \quad \text{(C.19)}$$

where $\boldsymbol{y}^j$'s are i.i.d. samples from p^*. The spectral norm of each term in the summation $\|S_k^{(l)}(\boldsymbol{y}^j_{C_k^{(l)}},:)^T T_k^{(l)}(\boldsymbol{y}^j_{C_k^{(l)}},:)\|_2$ is bounded by 1 since $S_k^{(l)}$ and $T_k^{(l)}$ are orthogonal matrices. Moreover, $\mathbb{E}[S_k^{(l)}(\boldsymbol{y}^j_{C_k^{(l)}},:)^T T_k^{(l)}(\boldsymbol{y}^j_{C_k^{(l)}},:)] = A_k^{(l)*}$ since $\mathbb{E}[\hat{p}] = p^*$. Therefore, we could use matrix Bernstein inequality in Corollary 10 and obtain

$$\mathbb{P}[\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_2 \geq t] \leq 2\tilde{r}^{(l)} \exp(\frac{-Nt^2/2}{1+2t/3}) \tag{C.20}$$

for any $t > 0$. We set $\tilde{\delta} = 2\tilde{r}^{(l)} \exp(\frac{-Nt^2/2}{1+2t/3})$ and suppose N is sufficient large such that $0 < \tilde{\delta} < 1$. Then $\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_2 < t$ holds with probability at least $1 - \tilde{\delta}$. We now get t in terms of $\tilde{\delta}$:

$$\begin{aligned} t &= \frac{-\frac{2}{3}\log(\frac{\tilde{\delta}}{2\tilde{r}^{(l)}}) + \sqrt{\frac{4}{9}\log^2(\frac{\tilde{\delta}}{2\tilde{r}^{(l)}}) - 2N\log(\frac{\tilde{\delta}}{2\tilde{r}^{(l)}})}}{N} = \frac{\frac{2}{3}\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}}) + \sqrt{\frac{4}{9}\log^2(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}}) + 2N\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}})}}{N} \\ &\leq \frac{\frac{2}{3}\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}}) + \frac{2}{3}\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}}) + \sqrt{2N\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}})}}{N} \leq \frac{(\frac{4}{3}+\sqrt{2})\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}})}{\sqrt{N}} \leq \frac{3\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}})}{\sqrt{N}}, \end{aligned} \tag{C.21}$$

where the first inequality follows $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b > 0$ and the second inequality follows the fact that $N > 1$ and $\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}}) > 1$ from $0 < \tilde{\delta} < 1$.

Therefore, using such a t in (C.21), we can have the following bound:

$$\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F \leq \sqrt{\tilde{r}^{(l)}}\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_2 \leq \frac{3\sqrt{\tilde{r}^{(l)}}\log(\frac{2\tilde{r}^{(l)}}{\tilde{\delta}})}{\sqrt{N}} \leq \frac{3\sqrt{\tilde{r}_{\texttt{max}}}\log(\frac{2\tilde{r}_{\texttt{max}}}{\tilde{\delta}})}{\sqrt{N}}, \tag{C.22}$$

which holds with probability at least $1 - \tilde{\delta}$ (here we recall notation $\tilde{r}_{\texttt{max}} = \max_l \tilde{r}^{(l)}$).

At this point we have a bound for $\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F$ for a specific k, l. However, we need a uniform bound over all possible choice of k, l where $k \in [2^l], l \in [L]$. There are at most

$d \log_2 d$ pairs of k, l, then

$$\begin{aligned}
&\mathbb{P}\left[\max_{k,l} \|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F \leq \frac{3\sqrt{\tilde{r}_{\mathtt{max}}}\log(\frac{2\tilde{r}_{\mathtt{max}}}{\tilde{\delta}})}{\sqrt{N}}\right] = 1 - \mathbb{P}\left[\max_{k,l} \|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F > \frac{3\sqrt{\tilde{r}_{\mathtt{max}}}\log(\frac{2\tilde{r}_{\mathtt{max}}}{\tilde{\delta}})}{\sqrt{N}}\right] \\
&\geq 1 - \mathbb{P}\left[\bigcup_{k,l}\left\{\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F > \frac{3\sqrt{\tilde{r}_{\mathtt{max}}}\log(\frac{2\tilde{r}_{\mathtt{max}}}{\tilde{\delta}})}{\sqrt{N}}\right\}\right] \\
&\geq 1 - \sum_{k,l}\mathbb{P}\left[\|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F > \frac{3\sqrt{\tilde{r}_{\mathtt{max}}}\log(\frac{2\tilde{r}_{\mathtt{max}}}{\tilde{\delta}})}{\sqrt{N}}\right] \geq 1 - \tilde{\delta} d \log_2 d. \qquad \text{(C.23)}
\end{aligned}$$

Let $\delta = \tilde{\delta} d \log_2 d$ and suppose N is sufficient large such that $0 < \delta < 1$. Thus, the following inequality for $\epsilon_A = \max_{k,l} \|\hat{A}_k^{(l)} - A_k^{(l)*}\|_F$ holds with probability at least $1 - \delta$:

$$\epsilon_A \leq \frac{3\sqrt{\tilde{r}_{\mathtt{max}}}\log(\frac{2\tilde{r}_{\mathtt{max}} d \log_2 d}{\delta})}{\sqrt{N}} \qquad \text{(C.24)}$$

□

APPENDIX D

APPENDIX OF OPTIMIZATION-FREE DIFFUSION MODEL

D.1 Coefficient Computation in High-dimensional Setting

In this appendix, we provide detailed computations of the high-dimensional coefficients involved in the linear system. Recall the definitions of the coefficient matrix $A(t) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ and the vector $\boldsymbol{b}^{(i)}(t) \in \mathbb{R}^{|\mathcal{S}|\times 1}$:

$$\begin{aligned} A_{l,l'}(t) &= \int \rho_t(\boldsymbol{x}) f_l(\boldsymbol{x}) f_{l'}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \\ b_l^{(i)}(t) &= \int \rho_t(\boldsymbol{x}) \left(\partial_{x_i} f_l(\boldsymbol{x}) - \beta f_l(\boldsymbol{x}) \partial_{x_i} V(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{D.1}$$

Each basis function $f_l(\boldsymbol{x})$ is of the form $\phi_{n_{k_1}}^{(k_1)}(x_{k_1})\phi_{n_{k_2}}^{(k_2)}(x_{k_2})$, for some pair of indices (k_1, k_2).

We compute $A(t)$ first. Using the eigenfunction property, we have:

$$A_{l,l'}(t) = \int \rho_t(\boldsymbol{x}) \left(\prod_{j=1}^{4} \phi_{n_{k_j}}^{(k_j)}(x_{k_j})\right) \mathrm{d}\boldsymbol{x} = \int \rho_0(\boldsymbol{x}) e^{\mathcal{L}t} \left(\prod_{j=1}^{4} \phi_{n_{k_j}}^{(k_j)}(x_{k_j})\right) \mathrm{d}\boldsymbol{x}$$

where the product arises from multiplying the two basis functions associated with indices l and l'. According to the local 2-cluster basis construction in equation (5.20), we have $k_1 \neq k_2$ and $k_3 \neq k_4$. Based on this structure, we categorize the computations into several distinct cases.

1. When all four indices are distinct ($k_1 \neq k_2 \neq k_3 \neq k_4$), the eigenfunctions are separable, and the operator $e^{\mathcal{L}t}$ can be applied to each term individually:

$$e^{\mathcal{L}t} \left(\prod_{j=1}^{4} \phi_{n_{k_j}}^{(k_j)}(x_{k_j})\right) = e^{(\sum_{j=1}^{4} \lambda_{n_{k_j}}^{(k_j)})t} \prod_{j=1}^{4} \phi_{n_{k_j}}^{(k_j)}(x_{k_j})$$

Therefore,

$$A_{l,l'}(t) = \int \rho_0(\boldsymbol{x}) e^{\mathcal{L}t} \left(\prod_{j=1}^{4} \phi_{n_{k_j}}^{(k_j)}(x_{k_j}) \right) \mathrm{d}\boldsymbol{x} = e^{(\sum_{j=1}^{4} \lambda_{n_{k_j}}^{(k_j)})t} \int \rho_0(\boldsymbol{x}) \prod_{j=1}^{4} \phi_{n_{k_j}}^{(k_j)}(x_{k_j}) \mathrm{d}\boldsymbol{x}$$

The integral on the right-hand side is evaluated using Monte Carlo integration with samples drawn from ρ_0 at the initial time.

2. When one pair among $\{x_{k_j}\}_{j=1}^{4}$ shares the same index (e.g., $k_1 = k_3 \neq k_2 \neq k_4$), we express the product of the two basis functions with the same variable as a linear combination in (5.14):
$\phi_{n_{k_1}}^{(k_1)}(x_{k_1})\phi_{n_{k_3}}^{(k_3)}(x_{k_3}) = \sum_{l_1=1}^{m} u_{l_1}^{(n_{k_1}, n_{k_3})} \phi_{l_1}^{(k_1)}(x_{k_1})$. Here $|\mathcal{S}'| = m$ follows the same expansion strategy demonstrated in Section 5.3.2. Therefore,

$$\begin{aligned}
&e^{\mathcal{L}t} \left(\prod_{j=1}^{4} \phi_{n_{k_j}}^{(k_j)}(x_{k_j}) \right) = e^{\mathcal{L}_{k_1} t} \left(\phi_{n_{k_1}}^{(k_1)}(x_{k_1}) \phi_{n_{k_3}}^{(k_3)}(x_{k_3}) \right) e^{\mathcal{L}_{k_2} t} \phi_{n_{k_2}}^{(k_2)}(x_{k_2}) e^{\mathcal{L}_{k_4} t} \phi_{n_{k_4}}^{(k_4)}(x_{k_4}) \\
&= e^{\mathcal{L}_{k_1} t} \left(\sum_{l_1=1}^{m} u_{l_1}^{(n_{k_1}, n_{k_2})} \phi_{l_1}^{(k_1)}(x_{k_1}) \right) e^{\lambda_{n_{k_3}}^{(k_3)} t} \phi_{n_{k_3}}^{(k_3)}(x_{k_3}) e^{\lambda_{n_{k_4}}^{(k_4)} t} \phi_{n_{k_4}}^{(k_4)}(x_{k_4}) \\
&= \left(\sum_{l_1=1}^{m} u_{l_1}^{(n_{k_1}, n_{k_2})} e^{\lambda_{l_1}^{(k_1)} t} \phi_{l_1}^{(k_1)}(x_{k_1}) \right) e^{\lambda_{n_{k_3}}^{(k_3)} t} \phi_{n_{k_3}}^{(k_3)}(x_{k_3}) e^{\lambda_{n_{k_4}}^{(k_4)} t} \phi_{n_{k_4}}^{(k_4)}(x_{k_4}) \\
&= \sum_{l_1=1}^{m} u_{l_1}^{(n_{k_1}, n_{k_2})} e^{(\lambda_{l_1}^{(k_1)} + \lambda_{k_3}^{(k_3)} + \lambda_{k_4}^{(k_4)})t} \phi_{l_1}^{(k_1)}(x_{k_1}) \phi_{n_{k_3}}^{(k_3)}(x_{k_3}) \phi_{n_{k_4}}^{(k_4)}(x_{k_4})
\end{aligned}$$

Hence,

$$A_{l,l'}(t) = \sum_{l_1=1}^{m} u_{l_1}^{(n_{k_1}, n_{k_2})} e^{(\lambda_{l_1}^{(k_1)} + \lambda_{k_3}^{(k_3)} + \lambda_{k_4}^{(k_4)})t} \int \rho_0(x) \phi_{l_1}^{(k_1)}(x_{k_1}) \phi_{n_{k_3}}^{(k_3)}(x_{k_3}) \phi_{n_{k_4}}^{(k_4)}(x_{k_4}) \mathrm{d}x$$

The integral on the right-hand side is evaluated using Monte Carlo integration, following the same approach as in the previous case.

3. When there are two repeated pairs among $\{x_{k_j}\}_{j=1}^4$ (e.g., $k_1 = k_3$, $k_2 = k_4$, with $k_1 \neq k_2$), we express each product of basis functions with repeated indices as a linear combination: $\phi_{n_{k_1}}^{(k_1)}(x_{k_1})\phi_{n_{k_3}}^{(k_3)}(x_{k_3}) = \sum_{l_1=1}^m u_{l_1}^{(n_{k_1},n_{k_3})}\phi_{l_1}^{(k_1)}(x_{k_1}), \phi_{n_{k_2}}^{(k_2)}(x_{k_2})\phi_{n_{k_4}}^{(k_4)}(x_{k_4}) = \sum_{l_1=1}^m u_{l_1}^{(n_{k_2},n_{k_4})}\phi_{l_1}^{(k_2)}(x_{k_2})$, and thus

$$
\begin{aligned}
e^{\mathcal{L}t}\left(\prod_{j=1}^4 \phi_{n_{k_j}}^{(k_j)}(x_{k_j})\right) &= e^{\mathcal{L}_{k_1}t}\left(\phi_{n_{k_1}}^{(k_1)}(x_{k_1})\phi_{n_{k_3}}^{(k_3)}(x_{k_3})\right) e^{\mathcal{L}_{k_2}t}\left(\phi_{n_{k_2}}^{(k_2)}(x_{k_2})\phi_{n_{k_4}}^{(k_4)}(x_{k_4})\right) \\
&= e^{\mathcal{L}_{k_1}t}\left(\sum_{l_1=1}^m u_{l_1}^{(n_{k_1},n_{k_3})}\phi_{l_1}^{(k_1)}(x_{k_1})\right) e^{\mathcal{L}_{k_2}t}\left(\sum_{l_2=1}^m u_{l_2}^{(n_{k_2},n_{k_4})}\phi_{l_2}^{(k_2)}(x_{k_2})\right) \\
&= \left(\sum_{l_1=1}^m u_{l_1}^{(n_{k_1},n_{k_3})} e^{\lambda_{l_1}^{(k_1)}t}\phi_{l_1}^{(k_1)}(x_{k_1})\right)\left(\sum_{l_2=1}^m u_{l_2}^{(n_{k_2},n_{k_4})} e^{\lambda_{l_2}^{(k_2)}t}\phi_{l_2}^{(k_2)}(x_{k_2})\right) \\
&= \sum_{l_1,l_2=1}^m u_{l_1}^{(n_{k_1},n_{k_3})} u_{l_2}^{(n_{k_2},n_{k_4})} e^{(\lambda_{l_1}^{(k_1)}+\lambda_{l_2}^{(k_2)})t}\phi_{l_1}^{(k_1)}(x_{k_1})\phi_{l_2}^{(k_2)}(x_{k_2})
\end{aligned}
$$

Therefore,

$$
A_{l,l'}(t) = \sum_{l_1,l_2=1}^m u_{l_1}^{(n_{k_1},n_{k_3})} u_{l_2}^{(n_{k_2},n_{k_4})} e^{(\lambda_{l_1}^{(k_1)}+\lambda_{l_2}^{(k_2)})t}\int \rho_0(x)\phi_{l_1}^{(k_1)}(x_{k_1})\phi_{l_2}^{(k_2)}(x_{k_2})\mathrm{d}x
$$

The case-by-case discussion above completes the computation of all entries in the matrix $A(t)$.

Next, we discuss the computation of $b^{(i)}(t)$. Since the chosen potential V is of mean-field form, the partial derivative $\partial_{x_i}V(\boldsymbol{x})$ depends only on the single variable x_i. Suppose we have the following linear expansions as in (5.14):

$$
\phi_{n_i}^{(i)'}(x_i) = \sum_{l_1=1}^m v_{l_1}^{(n_i)}\phi_{l_1}^{(i)}(x_i), \partial_{x_i}V(x) = \sum_{l_1=1}^m q_{l_1}^{(i)}\phi_{l_1}^{(i)}(x_i) \tag{D.2}
$$

Assume that the basis function $f_l(\boldsymbol{x})$ in equation (D.1) takes the form $f_l(\boldsymbol{x}) = \phi_{n_{k_1}}^{(k_1)}(x_{k_1})\phi_{n_{k_2}}^{(k_2)}(x_{k_2})$.

Then we have:

$$e^{\mathcal{L}t}\left(\partial_{x_i} f_l(\boldsymbol{x}) - \beta f_l(\boldsymbol{x})\partial_{x_i} V(\boldsymbol{x})\right) = e^{\mathcal{L}t}\left(\partial_{x_i}(\phi^{(k_1)}_{n_{k_1}}(x_{k_1})\phi^{(k_2)}_{n_{k_2}}(x_{k_2})) - \beta\phi^{(k_1)}_{n_{k_1}}(x_{k_1})\phi^{(k_2)}_{n_{k_2}}(x_{k_2})\partial_{x_i} V(\boldsymbol{x})\right)$$

We now compute the expression above on a case-by-case basis, depending on the relationship between i, k_1, and k_2.

1. If $k_1 \neq i$ and $k_2 \neq i$, then the derivative term $\partial_{x_i} f_l(\boldsymbol{x})$ vanishes. The expression simplifies to:

$$\begin{aligned}
&-\beta e^{\mathcal{L}t}\left(\phi^{(k_1)}_{n_{k_1}}(x_{k_1})\phi^{(k_2)}_{n_{k_2}}(x_{k_2})\partial_{x_i} V(\boldsymbol{x})\right) = -\beta e^{\mathcal{L}t}\left(\phi^{(k_1)}_{n_{k_1}}(x_{k_1})\phi^{(k_2)}_{n_{k_2}}(x_{k_2})\sum_{l_1=1}^{m} q^{(i)}_{l_1}\phi^{(i)}_{l_1}(x_i)\right) \\
&= -\beta\sum_{l_1=1}^{m} q^{(i)}_{l_1} e^{\mathcal{L}t}\left(\phi^{(k_1)}_{n_{k_1}}(x_{k_1})\phi^{(k_2)}_{n_{k_2}}(x_{k_2})\phi^{(i)}_{l_1}(x_i)\right) \\
&= -\beta\sum_{l_1=1}^{m} q^{(i)}_{l_1} e^{(\lambda^{(k_1)}_{n_{k_1}} + \lambda^{(k_2)}_{n_{k_2}} + \lambda^{(i)}_{l_1})t}\phi^{(k_1)}_{n_{k_1}}(x_{k_1})\phi^{(k_2)}_{n_{k_2}}(x_{k_2})\phi^{(i)}_{l_1}(x_i)
\end{aligned}$$

   Therefore,

$$b^{(i)}_l(t) = -\beta\sum_{l_1=1}^{m} q^{(i)}_{l_1} e^{(\lambda^{(k_1)}_{n_{k_1}} + \lambda^{(k_2)}_{n_{k_2}} + \lambda^{(i)}_{l_1})t}\int \rho_0(x)\phi^{(k_1)}_{n_{k_1}}(x_{k_1})\phi^{(k_2)}_{n_{k_2}}(x_{k_2})\phi^{(i)}_{l_1}(x_i)\mathrm{d}x$$

2. If $k_1 = i$ and $k_2 \neq k_1$, then the formula becomes

$$
\begin{aligned}
&e^{\mathcal{L}t}\left(\phi_{n_i}^{(i)\prime}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2}) - \beta\phi_{n_i}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2})\partial_{x_i}V(\boldsymbol{x})\right)\\
&= e^{\mathcal{L}t}\left((\sum_{l_1=1}^{m} v_{l_1}^{(n_i)}\phi_{l_1}^{(i)}(x_i))\phi_{n_{k_2}}^{(k_2)}(x_{k_2}) - \beta\phi_{n_i}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2})(\sum_{l_1=1}^{m} q_{l_1}^{(i)}\phi_{l_1}^{(i)}(x_i))\right)\\
&= \sum_{l_1=1}^{m} v_{l_1}^{(n_i)} e^{(\lambda_{l_1}^{(i)}+\lambda_{n_{k_2}}^{(k_2)})t}\phi_{l_1}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2}) - \beta\sum_{l_1=1}^{m} q_{l_1}^{(i)} e^{\mathcal{L}t}\left(\phi_{n_i}^{(i)}(x_i)\phi_{l_1}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2})\right)\\
&= \sum_{l_1=1}^{m} v_{l_1}^{(n_i)} e^{(\lambda_{l_1}^{(i)}+\lambda_{n_{k_2}}^{(k_2)})t}\phi_{l_1}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2}) - \beta\sum_{l_1=1}^{m} q_{l_1}^{(i)} e^{\mathcal{L}t}\left(\sum_{l_2=1}^{m} u_{l_2}^{(n_i,l_1)}\phi_{l_2}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2})\right)\\
&= \sum_{l_1=1}^{m} v_{l_1}^{(n_i)} e^{(\lambda_{l_1}^{(i)}+\lambda_{n_{k_2}}^{(k_2)})t}\phi_{l_1}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2}) - \beta\sum_{l_1=1}^{m} q_{l_1}^{(i)}\sum_{l_2=1}^{m} u_{l_2}^{(n_i,l_1)} e^{(\lambda_{l_2}^{(i)}+\lambda_{n_{k_2}}^{(k_2)})t}\left(\phi_{l_2}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2})\right)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
b_l^{(i)}(t) = &\sum_{l_1=1}^{m} v_{l_1}^{(n_i)} e^{(\lambda_{l_1}^{(i)}+\lambda_{n_{k_2}}^{(k_2)})t}\int \rho_0(x)\phi_{l_1}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2})\mathrm{d}x\\
&- \beta\sum_{l_2=1}^{m}\left(\sum_{l_1=1}^{m} q_{l_1}^{(i)} u_{l_2}^{(n_i,l_1)}\right) e^{(\lambda_{l_2}^{(i)}+\lambda_{n_{k_2}}^{(k_2)})t}\int \rho_0(x)\phi_{l_2}^{(i)}(x_i)\phi_{n_{k_2}}^{(k_2)}(x_{k_2})\mathrm{d}x
\end{aligned}
$$

3. If $k_2 = i$ and $k_2 \neq k_1$, the case is equivalent to the second case due to symmetry.

We summarize the computational complexity involved in computing $A(t)$ and $b^{(i)}(t)$. Note that there are $O(|\mathcal{S}|^2)$ integral terms of the form

$$\int \rho_0(\boldsymbol{x})\prod_{j=1}^{4}\phi_{n_{k_j}}^{(k_j)}(x_{k_j})\,\mathrm{d}\boldsymbol{x},$$

which dominate the cost of the initial Monte Carlo integration step. This results in a computational cost of $O(Nd_{\mathrm{b}}^2 d^2 n^4)$. Assume $m = O(n)$.

For the computation of $A(t)$:

- **Case 1**: There are $O(|\mathcal{S}|^2)$ terms, leading to a cost of $O(d_{\mathrm{b}}^2 d^2 n^4)$.
- **Case 2**: The number of relevant dimension index pairs is $O(d_{\mathrm{b}} d d_{\mathrm{b}})$. Taking into account the basis expansion, the total cost is $O(d_{\mathrm{b}} d d_{\mathrm{b}} m n^2) = O(d_{\mathrm{b}}^2 d n^3)$.
- **Case 3**: The total cost is $O(d_{\mathrm{b}} d m^2) = O(d_{\mathrm{b}} d n^2)$.

Summing over all cases, the overall computational complexity for $A(t)$ remains dominated by $O(d_{\mathrm{b}}^2 d^2 n^4)$.

For the computation of $b^{(i)}(t)$:

- **Case 1**: The cost is $O(d d_{\mathrm{b}} d n^2 m) = O(d_{\mathrm{b}} d^2 n^3)$.
- **Case 2**: The cost is $O(d d_{\mathrm{b}} n m^2) = O(d_{\mathrm{b}} d n^3)$.

As a result, the computational cost for evaluating $b^{(i)}(t)$ is smaller than that of $A(t)$. The total computational cost becomes $O(d_{\mathrm{b}}^2 d^2 n^4)$.

D.2 Example of Assumption

In this appendix, we provide two one-dimensional examples and one high-dimensional example to illustrate Assumption 7.

Example 4. *In the case of the Ornstein–Uhlenbeck process, the stationary distribution ρ_∞ is Gaussian. We choose f_n to be the eigenfunctions associated with ρ_∞, which correspond to Hermite polynomials H_n. Using this basis, we can expand the following two terms in terms of Hermite polynomials:*

$$e^{-a^2 x^2} = \sum_{i=0}^{\infty} \frac{(-1)^i a^{2i}}{i!(1+a^2)^{i+\frac{1}{2}} 2^{2i}} H_{2i}(x)$$

and

$$e^{bx} = e^{\frac{b^2}{4}} \sum_{i=0}^{\infty} \frac{b^i}{i! 2^i} H_i(x)$$

for some a and b. Besides, the ratio of two Gaussian distribution ρ_0/ρ_∞ is still a Gaussian distribution. Therefore, we can represent it by Hermite polynomials. Besides, when choosing appropriate δ, we have $\sup_{x\in\mathbb{R}} |\sum_{i=1}^{\infty} p_i f_i(x)| \leq 1$.

Next we verify the second assumption. For Hermite polynomial, we have $H_i'(x) = iH_{i-1}(x)$. We can also assume $|p_i| \leq \gamma^i$ for $\gamma < 1$ according to the previous expansion. Then we get

$$\nabla \sum_{i=M}^{\infty} p_i H_i(x) = \sum_{i=M}^{\infty} i p_i H_{i-1}(x)$$

and thus

$$\begin{aligned}
\int (\nabla \sum_{i=M}^{\infty} p_i H_i(x))^2 \rho_0(x) \mathrm{d}x \leq & C_\rho(1+\delta) \int (\sum_{i=M}^{\infty} i p_i H_{i-1}(x))^2 \rho_\infty(x) \mathrm{d}x \\
\leq & C_\rho(1+\delta) \sum_{i,j=M}^{\infty} ij|p_i||p_j| \int H_{i-1}(x) H_{j-1}(x) \rho_\infty(x) \mathrm{d}x \\
= & C_\rho(1+\delta) \sum_{i=M}^{\infty} i^2 |p_i|^2 \leq C_\rho(1+\delta) \sum_{i=M}^{\infty} i^2 \gamma^{2i} \\
\leq & C_\rho(1+\delta)\gamma^{2(M-1)} \left(\frac{(M-1)^2}{2\ln 1/\gamma} + \frac{M-1}{2\ln \gamma^2} + \frac{1}{4\ln \gamma^3} \right) =: r(M)
\end{aligned}$$

We denote the right hand side as $r(M)$. Due to $\gamma < 1$, the function converges to zero as $M \to \infty$.

Example 5. *In Fourier case defined in the domain $[-L, L]$, then ρ_∞ is a constant. We expand a Gaussian function $\rho_0 = \exp(-a^2x^2)$ as the following*

$$\rho_0 = C_\rho \rho_\infty (p_0 + \sum_{k=1}^{\infty} p_k \cos(\frac{\pi k x}{L}))$$

where $p_k = \frac{2}{C_\rho} \int_0^L e^{-a^2x^2} \cos(\frac{\pi k x}{L}) \mathrm{d}x \sim C \exp(-\frac{\pi k^2}{4a^2L^2})$ as $k \to \infty$, we can assume $|p_k| \leq \gamma^k$ for some $\gamma < 1$ when a is small. Thus, the condition $\sup_{x\in\mathbb{R}} |\sum_{i=1}^{\infty} p_i f_i(x)| \leq 1$ is

straightforward.

Besides, we follow the same logic as in the Hermite polynomial case and obtain

$$\begin{aligned}\int(\nabla\sum_{k=M}^{\infty}p_k\cos(\frac{\pi kx}{L}))^2\rho_0(x)\mathrm{d}x \leq& C_\rho(1+\delta)\int(\sum_{k=M}^{\infty}\frac{\pi k}{L}p_k\sin(\frac{\pi kx}{L}))^2\rho_\infty(x)\mathrm{d}x\\ \leq& C_\rho(1+\delta)\sum_{k,k'=M}^{\infty}\frac{\pi^2}{L^2}kk'|p_k||p_{k'}|\int\sin(\frac{\pi kx}{L})\sin(\frac{\pi k'x}{L})\rho_\infty(x)\mathrm{d}x\\ =& C_\rho(1+\delta)\sum_{k=M}^{\infty}\frac{\pi^2k^2}{L^2}|p_k|^2\leq C_\rho(1+\delta)\frac{\pi^2}{L^2}\sum_{k=M}^{\infty}k^2\gamma^{2k}\\ \leq& C_\rho(1+\delta)\frac{\pi^2}{L^2}\gamma^{2(M-1)}\left(\frac{(M-1)^2}{2\ln 1/\gamma}+\frac{M-1}{2\ln\gamma^2}+\frac{1}{4\ln\gamma^3}\right)=:r(M)\end{aligned}$$

Example 6. *We are in particular interested in high-dimensional potentials V that exhibit some "nearest-neighbor" interactions, commonly found in physics. For example let*

$$\begin{aligned}\rho_0&=\left(\prod_{i=1}^{d}\mu_i(x_i)\right)\exp\left(\delta\sum_{0<|i-j|\leq d_b}\alpha_{ij}x_ix_j\right)\\&=\left(\prod_{i=1}^{d}\mu_i(x_i)\right)\prod_{0<|i-j|\leq d_b}\exp\left(\delta\alpha_{ij}x_ix_j\right)\end{aligned}$$

where $\mu_i(x_i)$ is the mean-field term of the density and $\exp\left(\delta\alpha_{ij}x_ix_j\right)$ is the interaction term.

Suppose δ is small, which corresponds to a high-temperature scenario. We can expand $\exp(\delta\alpha_{ij}x_ix_j)$ in terms of δ:

$$\exp(\delta\alpha_{ij}x_ix_j)=1+\sum_{k=1}^{\infty}\delta^k\frac{\alpha_{ij}^k}{k!}x_i^rx_j^k=1+\delta\sum_{k=1}^{\infty}\delta^{k-1}\frac{\alpha_{ij}^k}{k!}x_i^kx_j^k.$$

Assume $\{f_{n_i}^{(i)}\}_{n\in\mathbb{N}}$ are the eigenfunctions associated to mean-field density $\mu_i(x_i)$. Now we

can expand $\sum_{k=1}^{\infty} \delta^{k-1} \frac{\alpha_{ij}^k}{k!} x_i^k x_j^k$ *in terms of eigenfunctions and thus, we have*

$$\exp(\delta \alpha_{ij} x_i x_j) = \left(1 + \delta \sum_{n_i, n_j = 1}^{\infty} \gamma_{n_i, n_j}^{i,j}(\delta) f_{n_i}^{(i)}(x_i) f_{n_j}^{(j)}(x_j) \right)$$

where we have that coefficient $\gamma_{n_i, n_j}^{i,j}(\delta) \to 0$ *as* $\delta \to 0$.

Let us call $\sum_{n_i, n_j = 1}^{\infty} \gamma_{n_i, n_j}^{i,j}(\delta) f_{n_i}^{(i)}(x_i) f_{n_j}^{(j)}(x_j) = C_2(i,j)$ *as a 2-cluster of pair* (i,j). *Then*

$$\exp(-\delta \alpha_{ij} x_i x_j) = 1 + \delta C_2(i,j)$$

and thus

$$\exp\left(\delta \sum_{i,j=1}^{d} \alpha_{ij} x_i x_j \right) = \prod_{i,j=1}^{d} (1 + \delta C_2(i,j)) = 1 + \delta\, C_2 + \delta^2 C_{>2}$$

Where C_2 *is the sum of all 2-cluster and this is also a 2-cluster, while* $C_{>2}$ *contains all products of 2-clusters and thus is a higher-order cluster.*

Therefore, we can interpret ρ_0 *as 2-cluster perturbation of* ρ_∞ *since*

$$\rho_0 = \rho_\infty (1 + \delta\, C_2 + \delta^2 C_{>2})$$

Based on Theorem 9 we can expand the score up to first order in δ *as a 2-cluster function.*

D.3 Proof of Theorem 10

In order to proof the theorem we need the following lemma.

Lemma 54. *Denote Fourier basis* $f_k(x) = e^{\sqrt{-1} w k x}$. *For* $\varepsilon > 0$ *there is* M *and* $g \in$

$\mathrm{span}\{f_{-M},\ldots,f_M\}$ *such that*

$$\left|\frac{1}{1+\delta\sum_{i=-\infty}^{\infty}p_ie^{\lambda_it}f_i(x)}-(1+g(x))\right|\leq\varepsilon e^{\lambda_1 t}$$

where we can choose $M\in\mathbb{N}$ *to satisfy*

$$\gamma^M\leq\frac{\varepsilon}{8\frac{\log(\frac{\varepsilon}{2}(1-\delta))}{\log\delta}}\frac{(1-\gamma)(1-\delta)}{\delta^2(1-\delta^{\frac{\log(\frac{\varepsilon}{2}(1-\delta))}{\log\delta}})}$$

as well as

$$(2M)^{\frac{\log(\frac{\varepsilon}{2}(1-\delta))}{\log\delta}}\gamma^M\leq\frac{\varepsilon}{8(1+\frac{\log(\frac{\varepsilon}{2}(1-\delta))}{\log\delta})}.$$

We put the proof of this lemma in the end. Now we go to Proof of Theorem 10 first.

Proof of Theorem 10. In the one-dimensional case,

$$\begin{aligned}\partial_x\log\rho_t(x)=&\partial_x\log(1+\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i(x))\\=&\frac{\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i'(x)}{1+\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i(x)}\end{aligned}$$

Therefore, for some $s_{2M}\in F_{2M}$, we have

$$\begin{aligned}&\int\frac{1}{2}(\partial_x\log\rho_t(x)-s_{2M}(t,x))^2\rho_t(x)\mathrm{d}x=\int(\frac{\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i'(x)}{1+\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i(x)}-s_{2M}(t,x))^2\rho_t(x)\mathrm{d}x\\&=\int\frac{1}{2}(\frac{\delta\sum_{i=M+1}^{\infty}p_ie^{\lambda_it}f_i'(x)}{1+\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i(x)}+\frac{\delta\sum_{i=1}^{M}p_ie^{\lambda_it}f_i'(x)}{1+\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i(x)}-s_{2M}(t,x))^2\rho_t(x)\mathrm{d}x\\&\leq\int(\frac{\delta\sum_{i=M+1}^{\infty}p_ie^{\lambda_it}f_i'(x)}{1+\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i(x)})^2\rho_t(x)+(\frac{\delta\sum_{i=1}^{M}p_ie^{\lambda_it}f_i'(x)}{1+\delta\sum_{i=1}^{\infty}p_ie^{\lambda_it}f_i(x)}-s_{2M}(t,x))^2\rho_t(x)\mathrm{d}x\end{aligned}$$

For the first term, note that $(\frac{1}{1+\delta\sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)})^2 \leq \frac{1}{(1-\delta)^2}$ and

$$\| \sum_{i=M+1}^{\infty} p_i f_i'(x)\|^2_{L^2(\rho_0)} \leq r(M+1)$$

shown in the assumption 7. Thus, there is a M such that

$$(\frac{\delta}{1-\delta})^2 \| \sum_{i=M+1}^{\infty} p_i f_i'(x)\|^2_{L^2(\rho_0)} \leq \frac{\varepsilon}{2}.$$

Therefore,

$$\int (\frac{\delta \sum_{i=M+1}^{\infty} p_i e^{\lambda_i t} f_i'(x)}{1+\delta\sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)})^2 \rho_t(x) \mathrm{d}x \leq \frac{\epsilon}{2} e^{2\lambda_{M+1} t}$$

For the second term, we follow Lemma 54 choosing $\tilde{\varepsilon} = \frac{\varepsilon}{2\|\sum_{i=1}^{\infty} p_i f_i'\|^2_{L^2(\rho_0)}}$ we have that there is $\tilde{s}_M \in \mathrm{span}\{f_{-M}, \ldots, f_M\}$ such that

$$\left| \frac{\delta \sum_{i=1}^{M} p_i e^{\lambda_i t} f_i'(x)}{1+\delta\sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)} - (\delta \sum_{i=1}^{M} p_i e^{\lambda_i t} f_i'(x))(1+\tilde{s}_M(t,x)) \right| \leq \frac{\varepsilon}{2} e^{\lambda_1 t}.$$

By observing that $f_n'(x) = \sqrt{-1} n\omega f_n(x)$ and $f_n(x) f_m(x) = f_{n+m}(x)$ we can expand

$$(\delta \sum_{i=1}^{M} p_i e^{\lambda_i t} f_i'(x))(1+\tilde{s}_M(t,x)) := s_{2M}(t,x)$$

in Fourier basis from $-2M$ to $2M$. Denote it as $s_{2M}(t,x)$. Therefore,

$$\int (\frac{\delta \sum_{i=1}^{M} p_i e^{\lambda_i t} f_i'(x)}{1+\delta\sum_{i=1}^{\infty} p_i e^{\lambda_i t} f_i(x)} - s_{2M}(t,x))^2 \rho_t(x) \mathrm{d}x \leq \frac{\varepsilon}{2} e^{\lambda_1 t}$$

In the end,

$$\int \frac{1}{2} (\partial_x \log \rho_t(x) - s_{2M}(t,x))^2 \rho_t(x) \mathrm{d}x \leq \frac{\epsilon}{2} e^{2\lambda_{M+1} t} + \frac{\varepsilon}{2} e^{\lambda_1 t} \leq \varepsilon e^{\lambda_1 t}$$

and

$$\int_0^\infty \int \frac{1}{2}(\partial_x \log \rho_t(x) - s_{2M}(t,x))^2 \rho_t(x) \mathrm{d}x\mathrm{d}t \leq \frac{\varepsilon}{|\lambda_1|}$$

□

Proof of Lemma 54. To ease the notation we assume $p_0 = 0$ according to our assumption. Let $\varepsilon > 0$. The proof of this theorem is based is on a more refined analysis of the expansion of

$$\frac{1}{1 + \sum_{i=-\infty}^{\infty} p_i e^{\lambda_i t} f_i(x)}.$$

Recall that $\frac{1}{1+x} = \sum_{k=0}^{\infty}(-1)^k x^k$ for $|x| < 1$. Thus, $\sum_{k=N_1}^{\infty} |x|^k \leq \frac{|x|^{N_1}}{1-|x|}$. With the same arguments as in the proof of Theorem 9 we can choose N_1 such that Let N_1 be such that

$$\frac{\delta^{N_1+1}}{1-\delta} \leq \frac{\varepsilon}{2} e^{\lambda_1 t}$$

We first split the geometric expansion as

$$\begin{aligned}\frac{1}{1 + \delta \sum_{i=-\infty}^{\infty} p_i e^{\lambda_i t} f_i(x)} = &\sum_{k=0}^{N_1} (-1)^k \left(\delta \sum_{i=-\infty}^{\infty} p_i e^{\lambda_i t} f_i(x) \right)^k \\ &+ \sum_{k=N_1+1}^{\infty} (-1)^k \left(\delta \sum_{i=-\infty}^{\infty} p_i e^{\lambda_i t} f_i(x) \right)^k .\end{aligned}$$

The second term is the truncation error of the geometric expansion. By choosing N_1 sufficiently large, this term can be made arbitrarily small.

We now focus on the finite sum. For each $1 \leq k \leq N_1$, we decompose the series into its

low-frequency and high-frequency parts:

$$\sum_{k=1}^{N_1}(-1)^k\left(\delta\sum_{i=-\infty}^{\infty}p_ie^{\lambda_i t}f_i(x)\right)^k$$
$$=\sum_{k=1}^{N_1}(-1)^k\left[\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_if_i(x)\right)+\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_if_i(x)\right)\right]^k.$$

Applying the binomial theorem gives

$$\sum_{k=1}^{N_1}(-1)^k\left(\delta\sum_{i=-\infty}^{\infty}p_ie^{\lambda_i t}f_i(x)\right)^k$$
$$=\sum_{k=1}^{N_1}(-1)^k\sum_{r=0}^{k}\binom{k}{r}\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_if_i(x)\right)^{k-r}\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_if_i(x)\right)^r$$
$$=\sum_{k=1}^{N_1}(-1)^k\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_if_i(x)\right)^k$$
$$+\sum_{k=1}^{N_1}(-1)^k\sum_{r=1}^{k}\binom{k}{r}\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_if_i(x)\right)^{k-r}\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_if_i(x)\right)^r.$$

The first term is the contribution from the truncated low-frequency expansion, while the second term contains at least one high-frequency factor.

We next estimate the second term. Since every term contains at least one copy of the

high-frequency tail, we factor out one such copy:

$$\begin{aligned}
&\sum_{k=1}^{N_1}(-1)^k\sum_{r=1}^{k}\binom{k}{r}\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_i f_i(x)\right)^{k-r}\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_i f_i(x)\right)^{r}\\
&=\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_i f_i(x)\right)\\
&\quad\cdot\sum_{k=1}^{N_1}(-1)^k\sum_{r=0}^{k-1}\binom{k}{r+1}\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_i f_i(x)\right)^{k-1-r}\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_i f_i(x)\right)^{r}.
\end{aligned}$$

Using

$$\binom{k}{r+1}=\frac{k}{r+1}\binom{k-1}{r},$$

we obtain

$$\begin{aligned}
&\left|\sum_{k=1}^{N_1}(-1)^k\sum_{r=1}^{k}\binom{k}{r}\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_i f_i(x)\right)^{k-r}\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_i f_i(x)\right)^{r}\right|\\
&\le\left|\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_i f_i(x)\right|\sum_{k=1}^{N_1}\sum_{r=0}^{k-1}\frac{k}{r+1}\binom{k-1}{r}\\
&\qquad\cdot\left(\sum_{i=-N_2}^{N_2}\delta|p_i|\right)^{k-1-r}\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta|p_i|\right)^{r}.
\end{aligned}$$

Since $k/(r+1)\le N_1$, the last expression is bounded by

$$\begin{aligned}
&N_1\left|\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_i f_i(x)\right|\sum_{k=1}^{N_1}\sum_{r=0}^{k-1}\binom{k-1}{r}\left(\sum_{i=-N_2}^{N_2}\delta|p_i|\right)^{k-1-r}\left(\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta|p_i|\right)^{r}\\
&=N_1\left|\sum_{i\in\mathbb{Z}\setminus[-N_2,N_2]}\delta e^{\lambda_i t}p_i f_i(x)\right|\sum_{k=1}^{N_1}\left(\sum_{i=-\infty}^{\infty}\delta|p_i|\right)^{k-1}.
\end{aligned}$$

Assuming $\sum_{i=-\infty}^{\infty} |p_i| \leq 1$, we have

$$\sum_{i=-\infty}^{\infty} \delta |p_i| \leq \delta.$$

Therefore,

$$\begin{aligned}
&\left| \sum_{k=1}^{N_1} (-1)^k \sum_{r=1}^{k} \binom{k}{r} \left(\sum_{i=-N_2}^{N_2} \delta e^{\lambda_i t} p_i f_i(x) \right)^{k-r} \left(\sum_{i \in \mathbb{Z} \setminus [-N_2, N_2]} \delta e^{\lambda_i t} p_i f_i(x) \right)^{r} \right| \\
&\leq N_1 \left| \sum_{i \in \mathbb{Z} \setminus [-N_2, N_2]} \delta e^{\lambda_i t} p_i f_i(x) \right| \frac{1 - \delta^{N_1}}{1 - \delta}.
\end{aligned}$$

Furthermore,

$$\left| \sum_{i \in \mathbb{Z} \setminus [-N_2, N_2]} \delta e^{\lambda_i t} p_i f_i(x) \right| \leq \delta e^{\lambda_1 t} \sum_{i \in \mathbb{Z} \setminus [-N_2, N_2]} |p_i|.$$

Thus

$$\begin{aligned}
&\left| \sum_{k=1}^{N_1} (-1)^k \sum_{r=1}^{k} \binom{k}{r} \left(\sum_{i=-N_2}^{N_2} \delta e^{\lambda_i t} p_i f_i(x) \right)^{k-r} \left(\sum_{i \in \mathbb{Z} \setminus [-N_2, N_2]} \delta e^{\lambda_i t} p_i f_i(x) \right)^{r} \right| \\
&\leq N_1 e^{\lambda_1 t} \frac{\delta(1 - \delta^{N_1})}{1 - \delta} \sum_{i \in \mathbb{Z} \setminus [-N_2, N_2]} |p_i| \\
&\leq 2 N_1 e^{\lambda_1 t} \frac{\delta(1 - \delta^{N_1})}{1 - \delta} \sum_{i=N_2}^{\infty} |p_i| \\
&\leq 2 N_1 e^{\lambda_1 t} \frac{\delta(1 - \delta^{N_1})}{1 - \delta} \frac{\gamma^{N_2}}{1 - \gamma}.
\end{aligned}$$

Next we expand the low-frequency term. By the multinomial theorem,

$$\sum_{k=1}^{N_1}(-1)^k\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_i f_i(x)\right)^k$$
$$=\sum_{k=1}^{N_1}(-\delta)^k\sum_{\substack{j_{-N_2},\dots,j_{N_2}\geq 0\\ \sum_{i=-N_2}^{N_2}j_i=k}}\binom{k}{j_{-N_2},\dots,j_{N_2}}\prod_{i=-N_2}^{N_2}\left(e^{\lambda_i t}p_i f_i(x)\right)^{j_i}.$$

Using the identity

$$\prod_{i=-N_2}^{N_2}f_i(x)^{j_i}=f_{\sum_{i=-N_2}^{N_2}ij_i}(x),$$

we can rewrite the above expression as

$$\sum_{k=1}^{N_1}(-\delta)^k\sum_{m=-kN_2}^{kN_2}f_m(x)\sum_{\substack{\sum_{i=-N_2}^{N_2}j_i=k\\ \sum_{i=-N_2}^{N_2}ij_i=m}}\binom{k}{j_{-N_2},\dots,j_{N_2}}\prod_{i=-N_2}^{N_2}e^{j_i\lambda_i t}p_i^{j_i}.$$

We now separate the modes $|m|\leq N_2$ from the modes $|m|>N_2$. This gives

$$\sum_{k=1}^{N_1}(-1)^k\left(\sum_{i=-N_2}^{N_2}\delta e^{\lambda_i t}p_i f_i(x)\right)^k$$
$$=\sum_{m=-N_2}^{N_2}f_m(x)\sum_{k=1}^{N_1}(-\delta)^k\sum_{\substack{\sum_{i=-N_2}^{N_2}j_i=k\\ \sum_{i=-N_2}^{N_2}ij_i=m}}\binom{k}{j_{-N_2},\dots,j_{N_2}}\prod_{i=-N_2}^{N_2}e^{j_i\lambda_i t}p_i^{j_i}$$
$$+\sum_{j=2}^{N_1}\sum_{m\in[-jN_2,jN_2]\setminus[-(j-1)N_2,(j-1)N_2]}f_m(x)\sum_{k=j}^{N_1}(-\delta)^k\sum_{\substack{\sum_{i=-N_2}^{N_2}j_i=k\\ \sum_{i=-N_2}^{N_2}ij_i=m}}\binom{k}{j_{-N_2},\dots,j_{N_2}}\prod_{i=-N_2}^{N_2}e^{j_i\lambda_i t}p_i^{j_i}.$$

The first term contains only the target modes $|m|\leq N_2$. The second term contains the

additional higher modes generated by multiplying the low-frequency functions.

We now bound this second contribution. Using $|f_m(x)| \le 1$, $|p_i| \le \gamma^{|i|}$, and the assumed bounds on the exponential factors, we obtain

$$\left| \sum_{j=2}^{N_1} \sum_{m \in [-jN_2, jN_2] \setminus [-(j-1)N_2, (j-1)N_2]} f_m(x) \sum_{k=j}^{N_1} (-\delta)^k \sum_{\substack{\sum_{i=-N_2}^{N_2} j_i = k \\ \sum_{i=-N_2}^{N_2} i j_i = m}} \binom{k}{j_{-N_2}, \ldots, j_{N_2}} \prod_{i=-N_2}^{N_2} e^{j_i \lambda_i t} p_i^{j_i} \right|$$

$$\le 2e^{\lambda_1 t} \sum_{j=2}^{N_1} \sum_{m=(j-1)N_2}^{jN_2} \gamma^m \sum_{k=j}^{N_1} \sum_{\substack{\sum_{i=-N_2}^{N_2} j_i = k \\ \sum_{i=-N_2}^{N_2} i j_i = m}} \binom{k}{j_{-N_2}, \ldots, j_{N_2}}.$$

Moreover,

$$\sum_{\substack{\sum_{i=-N_2}^{N_2} j_i = k \\ \sum_{i=-N_2}^{N_2} i j_i = m}} \binom{k}{j_{-N_2}, \ldots, j_{N_2}} \le (2N_2)^k.$$

Therefore,

$$\left| \sum_{j=2}^{N_1} \sum_{m \in [-jN_2, jN_2] \setminus [-(j-1)N_2, (j-1)N_2]} f_m(x) \sum_{k=j}^{N_1} (-\delta)^k \sum_{\substack{\sum_{i=-N_2}^{N_2} j_i = k \\ \sum_{i=-N_2}^{N_2} i j_i = m}} \binom{k}{j_{-N_2}, \ldots, j_{N_2}} \prod_{i=-N_2}^{N_2} e^{j_i \lambda_i t} p_i^{j_i} \right|$$

$$\le 2e^{\lambda_1 t} \sum_{j=2}^{N_1} \sum_{m=(j-1)N_2}^{jN_2} \gamma^m \sum_{k=j}^{N_1} (2N_2)^k$$

$$\le 2e^{\lambda_1 t} \sum_{j=2}^{N_1} \sum_{m=(j-1)N_2}^{jN_2} \gamma^m \frac{(2N_2)^{N_1}}{2N_2 - 1}$$

$$\le e^{\lambda_1 t} \frac{2(1+N_1)}{2N_2 - 1} (2N_2)^{N_1} \gamma^{N_2}.$$

Now choose N_2 sufficiently large so that

$$2N_1 \frac{\delta(1-\delta^{N_1})}{1-\delta} \frac{\gamma^{N_2}}{1-\gamma} \leq \frac{\varepsilon}{4}$$

and

$$\frac{2(1+N_1)}{2N_2-1}(2N_2)^{N_1}\gamma^{N_2} \leq \frac{\varepsilon}{4}.$$

This is possible because $0 < \gamma < 1$, so γ^{N_2} decays exponentially fast as $N_2 \to \infty$.

Define

$$\Pi_{N_2}(t,x) = 1 + \sum_{m=-N_2}^{N_2} f_m(x) \sum_{k=1}^{N_1} (-\delta)^k \sum_{\substack{\sum_{i=-N_2}^{N_2} j_i = k \\ \sum_{i=-N_2}^{N_2} i j_i = m}} \binom{k}{j_{-N_2}, \ldots, j_{N_2}} \prod_{i=-N_2}^{N_2} e^{j_i \lambda_i t} p_i^{j_i}.$$

Combining the geometric truncation error, the coefficient-tail error, and the high-frequency-mode error, we obtain

$$\left| \frac{1}{1+\delta \sum_{i=-\infty}^{\infty} p_i e^{\lambda_i t} f_i(x)} - \Pi_{N_2}(t,x) \right| \leq \varepsilon e^{\lambda_1 t}.$$

This completes the proof.

□

D.4 Proof of Theorem 11

We divide the proof into three steps. Step1 is aimed to show how to choose C_M in order to include the empirical coefficient $\hat{C}(t)$ in the space. Step2 is aimed to show how to choose C_M in order to include the constructed score coefficient (from approximation error) in the

space. Step3 is aimed to provide the upper bound of the statistical error.

1. **Step1**

Lemma 55. *Suppose* $\rho_0(x) = \rho_\infty(x)(1+\delta g(x)) = \rho_\infty(x)(1+\delta\sum_{i=1}^{\infty} p_i f_i(x))$, *which is a perturbation of mean-field density* ρ_∞, *then the following inequalities*

$$\frac{\|A(t)-A(\infty)\|_2}{\|A(\infty)\|_2} \leq \delta C_A, \frac{\|B(t)-B(\infty)\|_F}{\|B(\infty)\|_F} \leq \delta C_B \tag{D.3}$$

hold for some constant $C_A > 0$ *and* $C_B > 0$.

Proof. For A, the matrix elements are given by:

$$\begin{aligned}
&(A(t)-A(\infty))_{j,j'} = \int (\rho_t(x)-\rho_\infty(x)) f_j(x) f_{j'}(x) dx \\
&= \int e^{\mathcal{L}t}(f_j(x)f_{j'}(x))(\rho_0(x)-\rho_\infty(x))dx \\
&= \int e^{\mathcal{L}t}(f_j(x)f_{j'}(x))(\delta\rho_\infty(x)g(x))dx \\
&= \sum_{k\in\mathcal{S}} u_k^{(j,j')} e^{\lambda_k t} \int f_k(x)(\delta\rho_\infty(x)g(x))dx \\
&\leq \sqrt{\sum_{k\in\mathcal{S}}(u_k^{(j,j')})^2}\sqrt{\sum_{k\in\mathcal{S}} e^{2\lambda_k t}\delta^2(\int f_k(x)\rho_\infty(x)g(x)dx)^2}
\end{aligned} \tag{D.4}$$

Note that

$$\int f_k(x)\rho_\infty(x)g(x)dx = \int f_k(x)\rho_\infty(x)(\sum_{i=1}^{\infty} p_i f_i(x))dx = \sum_{i=1}^{\infty} p_i \int f_k(x)\rho_\infty(x)f_i(x)dx = p_k$$

due to f_i is orthogonal with respect to ρ_∞.

Therefore,

$$\|A(t)-A(\infty)\|_2 \le \|A(t)-A(\infty)\|_F = \sqrt{\sum_{j,j'}((A(t)-A(\infty))_{j,j'})^2}$$
$$= \sqrt{\left(\sum_{j,j'}\sum_{k\in\mathcal{S}}(u_k^{(j,j')})^2\right)\sum_{k\in\mathcal{S}} e^{2\lambda_k t}\delta^2 p_k^2}$$
$$\le \delta\sqrt{|\mathcal{S}|C_U^2\sum_{k\in\mathcal{S}} p_k^2} \tag{D.5}$$

Note that $\sum_{k\in\mathcal{S}} p_k^2 < \infty$ and $\|A(\infty)\|_2 = \|I\|_2 = 1$. Then $\frac{\|A(t)-A(\infty)\|_2}{\|A(\infty)\|_2} \le \delta C_A$ holds for some constant C_A.

Besides, for B,

$$\|B(t)-B(\infty)\|_F^2 = \sum_{j\in\mathcal{S}} \| \int (\rho_t(x)-\rho_\infty(x))(\nabla f_j(x) - f_j(x)\nabla V(x))dx\|_2^2$$
$$= \sum_{j\in\mathcal{S}} \| \int (\rho_t(x)-\rho_\infty(x))(\sum_{k\in\mathcal{S}} \tilde{q}_k^{(j)} f_k(x))dx\|_2^2$$
$$= \sum_{j\in\mathcal{S}} \| \int (\sum_{k\in\mathcal{S}} \tilde{q}_k^{(j)} e^{\lambda_k t} f_k(x))(\rho_0(x)-\rho_\infty(x))dx\|_2^2$$
$$\le \sum_{j\in\mathcal{S}}(\sum_{k\in\mathcal{S}}\|\tilde{q}_k^{(j)}\|_2^2)(\sum_{k\in\mathcal{S}} e^{2\lambda_k t}(\int f_k(x)(\delta\rho_\infty(x)g(x))dx)^2)$$
$$= \sum_{j\in\mathcal{S}}(\sum_{k\in\mathcal{S}}\|\tilde{q}_k^{(j)}\|_2^2)(\sum_{k\in\mathcal{S}} e^{2\lambda_k t}\delta^2 p_k^2) \le |\mathcal{S}|C_{\tilde{Q}}^2\|\tilde{Q}_1\|_F^2\delta^2\sum_{k\in\mathcal{S}} p_k^2 \tag{D.6}$$

Therefore,

$$\frac{\|B(t)-B(\infty)\|_F}{\|B(\infty)\|_F} \le \delta C_{\tilde{Q}}\sqrt{|\mathcal{S}|\sum_{k\in\mathcal{S}} p_k^2}$$

Thus, there exists constant $C_B > 0$ satisfying the condition.

□

Lemma 56. *Suppose* $|\int f_k(x)(\hat{\rho}_0(x)-\rho_0(x))dx| \leq \frac{C_f}{\sqrt{N}}$ *holds with constant* $C_f > 0$ *for any* $k \in \mathcal{S}$ *with high probability. Suppose the perturbation bound* $\frac{\|A(t)-A(\infty)\|_2}{\|A(\infty)\|_2} \leq \delta C_A$ *holds with some constant* $C_A > 0$ *for any* t *from Lemma 2. When* $\frac{C_U C_f |\mathcal{S}|}{\sqrt{N}} + \delta C_A < 1$, *there exists a constant* $C_M > 0$ *such that*

$$\|\hat{C}(t)\|_F \leq C_M$$

for any t. $\hat{C}(t)$ *is the empirical score coefficient.*

Proof. We aim to prove the bound $\|\hat{C}(t)\|_F \leq C_M$ for any t. We first establish bounds for the reference system, then extend to the perturbed system.

Note that

$$A(t)C(t) = -B(t).$$

When $t = \infty$, we have $A_{j,j'}(\infty) = \int \rho_\infty(x) f_j(x) f_{j'}(x) dx = \delta_{j,j'}$ then $A(\infty) = I$ as an identity matrix, and assume we represent $\nabla f_j(x) - f_j(x)\nabla V = \sum_{k\in\mathcal{S}} \tilde{q}_k^{(j)} f_k(x)$, then

$$B_j(\infty) = \int \rho_\infty(x)(\nabla f_j(x) - f_j(x)\nabla V)dx = \int \rho_\infty(x)(\sum_{k\in\mathcal{S}} \tilde{q}_k^{(j)} f_k(x))dx = \tilde{q}_1^{(j)}$$

Then

$$\|C(\infty)\|_F \leq \|B(\infty)\|_F \leq \|\tilde{Q}_1\|_F.$$

where $\tilde{Q}_1 = [\tilde{q}_1^{(1)}, \tilde{q}_1^{(2)}, \cdots, \tilde{q}_1^{(|\mathcal{S}|)}]^T$. Furthermore, assume the perturbations satisfy for general t:

$$\frac{\|A(t) - A(\infty)\|_2}{\|A(\infty)\|_2} \leq \delta C_A < 1, \quad \frac{\|B(t) - B(\infty)\|_F}{\|B(\infty)\|_F} \leq \delta C_B.$$

Note that $\hat{A}(t)\hat{C}(t) = -\hat{B}(t)$ and $A(\infty)C(\infty) = -B(\infty)$, we use the following pertur-

bation bound:

$$\frac{\|\hat{C}(t) - C(\infty)\|_F}{\|C(\infty)\|_F} \leq \frac{\kappa(A(\infty))}{1 - \kappa(A(\infty)) \cdot \frac{\|\hat{A}(t) - A(\infty)\|_2}{\|A(\infty)\|_2}} \cdot \left(\frac{\|\hat{A}(t) - A(\infty)\|_2}{\|A(\infty)\|_2} + \frac{\|\hat{B}(t) - B(\infty)\|_F}{\|B(\infty)\|_F} \right).$$

$\hat{A}(t), \hat{B}(t), \hat{C}(t)$ demonstrate that those matrices are computed from empirical distribution. We analyze the term $\|\hat{A}(t) - A(\infty)\|_2$ via:

$$\|\hat{A}(t) - A(\infty)\|_2 \leq \|\hat{A}(t) - A(t)\|_2 + \|A(t) - A(\infty)\|_2.$$

For the first term, the matrix elements are given by:

$$\begin{aligned}
&(\hat{A}(t) - A(t))_{j,j'} = \int (\hat{\rho}_t(x) - \rho_t(x)) f_j(x) f_{j'}(x) dx \\
&= \int e^{\mathcal{L}t} (f_j(x) f_{j'}(x)) (\hat{\rho}_0(x) - \rho_0(x)) dx \\
&= \sum_{k \in \mathcal{S}} u_k^{(j,j')} e^{\lambda_k t} \int f_k(x) (\hat{\rho}_0(x) - \rho_0(x)) dx \\
&\leq \sqrt{\sum_{k \in \mathcal{S}} (u_k^{(j,j')})^2} \sqrt{\sum_{k \in \mathcal{S}} e^{2\lambda_k t} (\int f_k(x) (\hat{\rho}_0(x) - \rho_0(x)) dx)^2}
\end{aligned} \tag{D.7}$$

where we decompose $f_j(x) f_{j'}(x) = \sum_{k \in \mathcal{S}} u_k^{(j,j')} f_k(x)$. Suppose $|\int f_k(x) (\hat{\rho}_0(x) - \rho_0(x)) dx| \leq \frac{C_f}{\sqrt{N}}$ for any k w.h.p. and denote $C_U = \max_k \|U_k\|_F$.

Thus,

$$\|\hat{A}(t) - A(t)\|_2 \leq \|\hat{A}(t) - A(t)\|_F = \sqrt{\sum_{j,j'} [(\hat{A}(t) - A(t))_{j,j'}]^2} \leq \sqrt{(\sum_{j,j'} \sum_{k \in \mathcal{S}} (u_k^{(j,j')})^2) |\mathcal{S}| \frac{C_f^2}{N} e^{2\lambda_1 t}}$$

$$\leq \sqrt{|\mathcal{S}| C_U^2 |\mathcal{S}| \frac{C_f^2 e^{2\lambda_1 t}}{N}} = \frac{C_U C_f |\mathcal{S}|}{\sqrt{N}} e^{\lambda_1 t} \tag{D.8}$$

Then including the second term, we obtain:

$$\|\hat{A}(t) - A(\infty)\|_2 \leq \frac{C_U C_f |\mathcal{S}|}{\sqrt{N}} e^{\lambda_1 t} + \delta C_A.$$

Similarly, we decompose the error for B:

$$\|\hat{B}(t) - B(t)\|_F \leq \|\hat{B}(t) - B(t)\|_F + \|B(t) - B(\infty)\|_F.$$

The first term can be bounded as:

$$\begin{aligned}
\|\hat{B}(t) - B(t)\|_F^2 &= \sum_{j\in\mathcal{S}} \| \int (\hat{\rho}_t(x) - \rho_t(x))(\nabla f_j(x) - f_j(x)\nabla V(x))dx\|_2^2 \\
&= \sum_{j\in\mathcal{S}} \| \int (\hat{\rho}_t(x) - \rho_t(x))(\sum_{k\in\mathcal{S}} \tilde{q}_k^{(j)} f_k(x))dx\|_2^2 \\
&= \sum_{j\in\mathcal{S}} \| \int (\sum_{k\in\mathcal{S}} \tilde{q}_k^{(j)} e^{\lambda_k t} f_k(x))(\hat{\rho}_0(x) - \rho_0(x))dx\|_2^2 \\
&\leq \sum_{j\in\mathcal{S}} (\sum_{k\in\mathcal{S}} \|\tilde{q}_k^{(j)}\|_2^2)(\sum_{k\in\mathcal{S}} e^{2\lambda_k t} (\int f_k(x)(\hat{\rho}_0(x) - \rho_0(x))dx)^2)
\end{aligned} \tag{D.9}$$

Then

$$\|\hat{B}(t) - B(t)\|_F \leq \sqrt{|\mathcal{S}|} C_{\tilde{Q}} \frac{C_f \sqrt{|\mathcal{S}|}}{\sqrt{N}} e^{\lambda_1 t} = \frac{C_{\tilde{Q}} C_f |\mathcal{S}|}{\sqrt{N}} e^{\lambda_1 t},$$

where $C_{\tilde{Q}} = \max_k \|\tilde{Q}_k\|_F / \|B(\infty)\|_F = \max_k \|\tilde{Q}_k\|_F / \|\tilde{Q}_1\|_F$.

Thus,

$$\|\hat{B}(t) - B(\infty)\|_F \leq \left(\frac{C_{\tilde{Q}} C_f |\mathcal{S}|}{\sqrt{N}} e^{\lambda_1 t} + \delta C_B \right) \|B(\infty)\|_F.$$

Combining all results, we receive the perturbation bound:

$$\frac{\|\hat{C}(t) - C(\infty)\|_F}{\|C(\infty)\|_F} \leq \frac{1}{1 - \left(\frac{C_U C_f |\mathcal{S}|}{\sqrt{N}} + \delta C_A \right)} \cdot \left(\frac{C_U C_f |\mathcal{S}|}{\sqrt{N}} + \delta C_A + \frac{C_{\tilde{Q}} C_f |\mathcal{S}|}{\sqrt{N}} + \delta C_B \right).$$

The final bound becomes:

$$\|\hat{C}(t)\|_F \leq \frac{1+\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_B}{1-\left(\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_A\right)}\|C(\infty)\|_F = \frac{1+\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_B}{1-\left(\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_A\right)}\|\tilde{Q}_1\|_F,$$

under the assumption:

$$\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_A < 1.$$

Here we take

$$C_M = \frac{1+\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_B}{1-\left(\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_A\right)}\|C(\infty)\|_F = \frac{1+\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_B}{1-\left(\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}+\delta C_A\right)}\|\tilde{Q}_1\|_F$$

and receive the norm upper bound $\|\hat{C}(t)\|_F \leq C_M$ for any t. □

2. **Step2**

The next part is to verify the constructed score function belongs to the considered space. Recall the construction score for the approximation error:

$$s_{//}(t,x) = \nabla V(x) + \delta \sum_{i=1}^{M} p_i e^{\lambda_i t} \nabla f_i(x) = \nabla V(x) + \delta \sum_{i=1}^{M} p_i e^{\lambda_i t} (\sum_{j\in\mathcal{S}} v_j^{(i)} f_j(x)) \quad \text{(D.10)}$$

Then the norm of coefficient is upper bounded:

$$\|C_{//}(t)\|_2^2 \leq \delta^2 \sum_{j\in\mathcal{S}} \| \sum_{i=1}^{M} p_i e^{\lambda_i t} v_j^{(i)}\|_2^2 \leq C_M^2$$

for some sufficient small δ.

3. **Step3**

Given $\|\hat{C}(t)\|_F \leq C_M$ for all t, we aim to estimate the statistical error:

$$\text{err}_{\text{stat}} = \left| \iint \left(\frac{1}{2} \|s(t,x)\|^2 + \text{div}(s(t,x)) \right) (\rho_t(x) - \hat{\rho}_t(x))\, dx\, dt \right|,$$

where

$$s^{(i)}(t,x) = \langle c^{(i)}(t), f(x) \rangle - \partial_{x_i} V(x).$$

We define a space:

$$\mathcal{H}_S = \{ s^{(i)}(t,x) = \langle c^{(i)}(t), f(x) \rangle - \partial_{x_i} V(x), c^{(i)}(t) \in \mathbb{R}^{|\mathcal{S}| \times 1}, \|C(t)\|_F \leq C_M, \text{for all} \quad t \}$$

We give the statistical error for all score functions belonging to the above set.

Expanding the expression:

$$\begin{aligned}
\text{err}_{\text{stat}} = & \left| \iint \left[\frac{1}{2} \sum_{i=1}^{d} (\langle c^{(i)}, f \rangle)^2 - \sum_{i=1}^{d} \langle c^{(i)}, f \rangle \partial_{x_i} V + \frac{1}{2} \sum_{i=1}^{d} (\partial_{x_i} V)^2 \right. \right. \\
& \left. \left. + \sum_{i=1}^{d} \langle c^{(i)}, \partial_{x_i} f \rangle - \sum_{i=1}^{d} \partial_{x_i}^2 V \right] (\rho_t(x) - \hat{\rho}_t(x))\, dx\, dt \right| \\
\leq & \left| \iint \frac{1}{2} \sum_{i=1}^{d} (\langle c^{(i)}, f \rangle)^2 (\rho_t(x) - \hat{\rho}_t(x))\, dx\, dt \right| \\
& + \left| \iint \sum_{i=1}^{d} \langle c^{(i)}, \partial_{x_i} f - f \partial_{x_i} V \rangle (\rho_t(x) - \hat{\rho}_t(x))\, dx\, dt \right| \\
& + \left| \iint \sum_{i=1}^{d} \left(\frac{1}{2} (\partial_{x_i} V)^2 - \partial_{x_i}^2 V \right) (\rho_t(x) - \hat{\rho}_t(x))\, dx\, dt \right|.
\end{aligned}$$

We denote these three terms as (1), (2), and (3) respectively.

Bound for Term (1):

$$(1) = \frac{1}{2}\left|\iint \sum_{i=1}^{d} \|\langle c^{(i)}, f\rangle\|^2(\rho_t(x) - \hat{\rho}_t(x))\,dx\,dt\right|$$
$$= \frac{1}{2}\left|\iint \sum_{i=1}^{d}\sum_{j,j'} c_j^{(i)}(t)c_{j'}^{(i)}(t)f_j(x)f_{j'}(x)(\rho_t(x) - \hat{\rho}_t(x))\,dx\,dt\right|.$$

For fixed t, we analyze the matrix expression:

$$\sum_{i=1}^{d}\sum_{j,j'} c_j^{(i)}(t)c_{j'}^{(i)}(t)[A_{jj'}(t) - \hat{A}_{jj'}(t)] = \sum_{i=1}^{d}(c^{(i)}(t))^\top[A(t) - \hat{A}(t)]c^{(i)}(t)$$
$$\leq \sum_{i=1}^{d}\|c^{(i)}(t)\|_2^2 \cdot \|A(t) - \hat{A}(t)\|_2 \leq \|C\|_F^2 \cdot \|A(t) - \hat{A}(t)\|_2 \leq C_M^2\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}e^{\lambda_1 t}.$$

Therefore, by crossing all the time t,

$$(1) \leq \frac{1}{2}C_M^2\frac{C_U C_f|\mathcal{S}|}{\sqrt{N}}\frac{1}{|\lambda_1|}.$$

Bound for Term (2):

$$(2) = \left|\iint \sum_{i=1}^{d}\langle c^{(i)}(t), \partial_{x_i}f - f\partial_{x_i}V\rangle(\rho_t(x) - \hat{\rho}_t(x))\,dx\,dt\right|$$
$$\leq \sqrt{\sum_{i=1}^{d}\|c^{(i)}(t)\|_2^2} \cdot \sqrt{\sum_{i=1}^{d}\sum_{k\in\mathcal{S}}\left(\int(\partial_{x_i}f_k - f_k\partial_{x_i}V)(\rho_t(x) - \hat{\rho}_t(x))\,dx\right)^2}$$
$$\leq C_M\|B(t) - \hat{B}(t)\|_F$$
$$\leq C_M\frac{C_{\tilde{Q}}C_f|\mathcal{S}|}{\sqrt{N}}e^{\lambda_1 t}.$$

Therefore,

$$(2) \le C_M \frac{C_{\tilde{Q}} C_f |\mathcal{S}|}{\sqrt{N}} \frac{1}{|\lambda_1|}.$$

Bound for Term (3):

$$\begin{aligned}
(3) &= \left| \iint \sum_{i=1}^{d} \left(\frac{1}{2} (\partial_{x_i} V)^2 - \partial_{x_i}^2 V \right) (\rho_t(x) - \hat{\rho}_t(x))\, dx\, dt \right| \\
&= \left| \iint \sum_{i=1}^{d} \sum_{k \in \mathcal{S}} v_k^{(i)} f_k(x) (\rho_t(x) - \hat{\rho}_t(x))\, dx\, dt \right| \\
&\le \int \sqrt{\sum_{k \in \mathcal{S}} \left(\sum_{i=1}^{d} v_k^{(i)} \right)^2} \cdot \sqrt{\sum_{k \in \mathcal{S}} \left(\int f_k(x) (\rho_t(x) - \hat{\rho}_t(x))\, dx \right)^2} dt \\
&\le \cdot \sqrt{d} C_V \cdot \sqrt{|\mathcal{S}|} \cdot \frac{C_f}{\sqrt{N}} \frac{1}{|\lambda_1|} \\
&= \frac{C_V C_f \sqrt{d} \sqrt{|\mathcal{S}|}}{\sqrt{N}} \frac{1}{|\lambda_1|}.
\end{aligned}$$

Combining all three terms:

$$\begin{aligned}
\mathrm{err}_{\mathrm{stat}} &\le \frac{1}{2} C_M^2 \frac{C_U C_f |\mathcal{S}|}{\sqrt{N}} \frac{1}{|\lambda_1|} + C_M \frac{C_{\tilde{Q}} C_f |\mathcal{S}|}{\sqrt{N}} \frac{1}{|\lambda_1|} + \frac{C_V C_f \sqrt{d} \sqrt{|\mathcal{S}|}}{\sqrt{N}} \frac{1}{|\lambda_1|} \\
&= \left(\frac{1}{2} C_M^2 C_U C_f |\mathcal{S}| + C_M C_{\tilde{Q}} C_f |\mathcal{S}| + C_V C_f \sqrt{d} \sqrt{|\mathcal{S}|} \right) \frac{1}{|\lambda_1| \sqrt{N}}.
\end{aligned}$$

4. Additional lemma

Lemma 57 (Perturbation of a linear system)**.** *Let $A \in \mathbb{R}^{n \times n}$ be nonsingular and $X, B \in \mathbb{R}^{n \times d}$ satisfy $AX = B$. Let $\Delta A, \Delta B$ be perturbations, and let $X + \Delta X$ solve*

$$(A + \Delta A)(X + \Delta X) = B + \Delta B.$$

Assume a submultiplicative matrix norm $\|\cdot\|$ and that $\|A^{-1}\Delta A\| < 1$. Then

$$\|\Delta X\| \;\le\; \frac{\|A^{-1}\|}{1-\|A^{-1}\Delta A\|}\Big(\,\|\Delta B\| + \|\Delta A\|\,\|X\|\,\Big).$$

Moreover, using $\kappa(A) := \|A\|\,\|A^{-1}\|$ and $\|B\| = \|AX\| \le \|A\|\,\|X\|$,

$$\frac{\|\Delta X\|}{\|X\|} \;\le\; \frac{\kappa(A)}{1-\|A^{-1}\Delta A\|}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta B\|}{\|B\|}\right).$$

Proof. From $(A+\Delta A)(X+\Delta X) = B + \Delta B$ and $AX = B$,

$$(A+\Delta A)\,\Delta X = \Delta B - \Delta A\,X, \qquad \Delta X = (A+\Delta A)^{-1}(\Delta B - \Delta A\,X).$$

Write $(A+\Delta A)^{-1} = (I + A^{-1}\Delta A)^{-1}A^{-1}$. If $\|A^{-1}\Delta A\| < 1$, then $\|(I+A^{-1}\Delta A)^{-1}\| \le 1/(1-\|A^{-1}\Delta A\|)$ (Neumann series). Submultiplicativity gives

$$\|\Delta X\| \le \frac{\|A^{-1}\|}{1-\|A^{-1}\Delta A\|}\big(\|\Delta B\| + \|\Delta A\|\,\|X\|\big),$$

and the relative bound follows by dividing by $\|X\|$ and using $\|B\| \le \|A\|\,\|X\|$. $\square$

REFERENCES

[1] Salman Ahmadi-Asl, Stanislav Abukhovich, Maame G Asante-Mensah, Andrzej Cichocki, Anh Huy Phan, Tohishisa Tanaka, and Ivan Oseledets. Randomized algorithms for computation of tucker decomposition and higher order svd (hosvd). *IEEE Access*, 9:28684–28706, 2021.

[2] Talal Ahmed, Haroon Raja, and Waheed U. Bajwa. Tensor regression using low-rank and sparse tucker decompositions. *SIAM Journal on Mathematics of Data Science*, 2(4):944–966, 2020.

[3] Hussam Al Daas, Grey Ballard, Paul Cazeaux, Eric Hallman, Agnieszka Miedlar, Mirjeta Pasha, Tim W Reid, and Arvind K Saibaba. Randomized algorithms for rounding in the tensor-train format. *SIAM Journal on Scientific Computing*, 45(1):A74–A95, 2023.

[4] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. *Advances in neural information processing systems*, 28, 2015.

[5] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[6] Michael S Albergo, Mark Goldstein, Nicholas M Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. *arXiv preprint arXiv:2310.03725*, 2023.

[7] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[8] Sivaram Ambikasaran, Daniel Foreman-Mackey, Leslie Greengard, David W Hogg, and Michael O'Neil. Fast direct methods for gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):252–265, 2015.

[9] Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[10] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *International Conference on Machine Learning*, 2017.

[11] Francis Bach. *Learning theory from first principles*. MIT press, 2024.

[12] Markus Bachmayr, Reinhold Schneider, and André Uschmajew. Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations. *Foundations of Computational Mathematics*, 16:1423–1472, 2016.

[13] Roland Badeau and Rémy Boyer. Fast multilinear singular value decomposition for structured tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1008–1021, 2008.

[14] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*, 2015.

[15] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 843–852, 2023.

[16] Mario Bebendorf. *Hierarchical matrices*. Springer, 2008.

[17] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[18] Adam Block, Youssef Mroueh, and Alexander Rakhlin. Generative modeling with denoising auto-encoders and langevin sampling. *arXiv preprint arXiv:2002.00107*, 2020.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[20] Tai-Danae Bradley, E Miles Stoudenmire, and John Terilla. Modeling sequences with quantum states: a look under the hood. *Machine Learning: Science and Technology*, 1(3):035008, 2020.

[21] Tai-Danae Bradley, E Miles Stoudenmire, and John Terilla. Modeling sequences with quantum states: a look under the hood. *Machine Learning: Science and Technology*, 1(3):035008, 2020.

[22] H Brezis. Functional analysis, sobolev spaces and partial differential equations, 2011.

[23] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[24] T Tony Cai and Anru Zhang. Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics. 2018.

[25] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[26] Jian Cao, Marc G Genton, David E Keyes, and George M Turkiyyah. Hierarchical-block conditioning approximations for high-dimensional multivariate normal probabilities. *Statistics and Computing*, 29(3):585–598, 2019.

[27] 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.

[28] Stanley Chan et al. Tutorial on diffusion models for imaging and vision. *Foundations and Trends® in Computer Graphics and Vision*, 16(4):322–471, 2024.

[29] Kamalika Chaudhuri, Sanjoy Dasgupta, Samory Kpotufe, and Ulrike Von Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912, 2014.

[30] Maolin Che and Yimin Wei. Randomized algorithms for the approximations of tucker and the tensor train decompositions. *Advances in Computational Mathematics*, 45(1):395–428, 2019.

[31] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023.

[32] Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization. *arXiv preprint arXiv:2404.07771*, 2024.

[33] Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1(1):161–187, 2017.

[34] Yian Chen and Mihai Anitescu. Scalable gaussian process analysis for implicit physics-based covariance models. *International Journal for Uncertainty Quantification*, 11(6), 2021.

[35] Yian Chen and Mihai Anitescu. Scalable physics-based maximum likelihood estimation using hierarchical matrices. *arXiv preprint arXiv:2303.10102*, 2023.

[36] Yian Chen, Jeremy Hoskins, Yuehaw Khoo, and Michael Lindsey. Committor functions via tensor networks. *Journal of Computational Physics*, 472:111646, 2023.

[37] Yian Chen and Yuehaw Khoo. Combining particle and tensor-network methods for partial differential equations via sketching. *arXiv preprint arXiv:2305.17884*, 2023.

[38] Yian Chen, Yuehaw Khoo, and Michael Lindsey. Multiscale semidefinite programming approach to positioning problems with pairwise structure. *arXiv preprint arXiv:2012.10046*, 2020.

[39] Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, et al. Spectral methods for data science: A statistical perspective. *Foundations and Trends® in Machine Learning*, 14(5):566–806, 2021.

[40] Mathilde Chevreuil, Régis Lebrun, Anthony Nouy, and Prashant Rai. A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):897–921, 2015.

[41] Andrzej Cichocki, Danilo P Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar F Caiafa, and Anh Huy Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[42] Pierre Comon, Xavier Luciani, and André LF De Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 23(7-8):393–405, 2009.

[43] Kyle Cranmer. Kernel estimation in high-energy physics. *Computer Physics Communications*, 136(3):198–207, 2001.

[44] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero

Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

[45] Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. Remarks on some nonparametric estimates of a density function. *Selected Works of Murray Rosenblatt*, pages 95–100, 2011.

[46] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[47] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[48] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r1, r2,..., rn) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[49] Arthur Dempster. Maximum likelihood estimation from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[50] Ronald A DeVore and George G Lorentz. *Constructive approximation*, volume 303. Springer Science & Business Media, 1993.

[51] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 2021.

[52] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[53] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[54] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[55] Sergey Dolgov, Karim Anaya-Izquierdo, Colin Fox, and Robert Scheichl. Approximation and sampling of multivariate probability distributions in the tensor train decomposition. *Statistics and Computing*, 30:603–625, 2020.

[56] Sergey V. Dolgov and Boris N. Khoromskij. Simultaneous state-time approximation of the chemical master equation using tensor product formats. *Numerical Linear Algebra with Applications*, 22(2):197–219, 2015.

[57] Sergey V Dolgov and Dmitry V Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM Journal on Scientific Computing*, 36(5):A2248–A2271, 2014.

[58] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B*, 99(1):014104, 2019.

[59] Petros Drineas and Michael W Mahoney. Randnla: randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

[60] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

[61] Genevieve Dusson, Markus Bachmayr, Gábor Csányi, Ralf Drautz, Simon Etter, Cas van der Oord, and Christoph Ortner. Atomic cluster expansion: Completeness, efficiency and stability. *Journal of Computational Physics*, 454:110946, 2022.

[62] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[63] Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577–588, 1995.

[64] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[65] Walter Gautschi. *Orthogonal polynomials: computation and approximation*. OUP Oxford, 2004.

[66] Marc G Genton, David E Keyes, and George Turkiyyah. Hierarchical decompositions for the computation of high-dimensional multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 27(2):268–277, 2018.

[67] Christopher J Geoga, Mihai Anitescu, and Michael L Stein. Scalable gaussian process computations using hierarchical matrices. *Journal of Computational and Graphical Statistics*, 29(2):227–237, 2020.

[68] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.

[69] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.

[70] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[71] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[72] Lars Grasedyck. Hierarchical tucker decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[73] Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[74] Lars Grasedyck, Daniel Kressner, and Christine Tobler. Alternating least squares tensor completion in the tt format. *SIAM Journal on Scientific Computing*, 37(1):A242–A263, 2015.

[75] Karl Gregory, Enno Mammen, and Martin Wahl. Statistical inference in sparse high-dimensional additive models. *The Annals of Statistics*, 49(3):1514–1536, 2021.

[76] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[77] Rajarshi Guhaniyogi, Shaan Qamar, and David B. Dunson. Bayesian tensor regression. *Journal of Machine Learning Research*, 18(79):1–31, 2017.

[78] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[79] Wolfgang Hackbusch. A sparse matrix arithmetic based on-matrices. part i: Introduction to-matrices. *Computing*, 62(2):89–108, 1999.

[80] Wolfgang Hackbusch and Boris N Khoromskij. A sparse h-matrix arithmetic. part ii: Application to multi-dimensional problems. *Computing*, 64:21–47, 2000.

[81] Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, 2009.

[82] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[83] Yinbin Han, Meisam Razaviyayn, and Renyuan Xu. Neural network-based score estimation in diffusion models: Optimization and generalization. *arXiv preprint arXiv:2401.15604*, 2024.

[84] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3):031012, 2018.

[85] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3):031012, 2018.

[86] Botao Hao, Boxiang Wang, Pengyuan Wang, Jingfei Zhang, Jian Yang, and Will Wei Sun. Sparse tensor additive regression. *Journal of Machine Learning Research*, 22(64):1–43, 2021.

[87] Asad Haris, Noah Simon, and Ali Shojaie. Generalized sparse additive models. *Journal of Machine Learning Research*, 23(70):1–56, 2022.

[88] Richard A Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[89] Amelia Henriksen and Rachel Ward. Adaoja: Adaptive learning rates for streaming pca. *arXiv preprint arXiv:1905.12115*, 2019.

[90] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

[91] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[92] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[93] Peter D. Hoff. Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics*, 9(3):1169–1193, 2015.

[94] Karl-Heinz Hoffmann and Qi Tang. *Ginzburg-Landau Phase Transition Theory and Superconductivity*. Birkhäuser Basel, 2012.

[95] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. Manifolds of tensors of fixed tt-rank. *Numerische Mathematik*, 120(4):701–731, 2012.

[96] Roger A Horn and Charles R Johnson. *Topics in matrix analysis*. Cambridge university press, 1994.

[97] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[98] Jeremy Hoskins, Yuehaw Khoo, Oscar Mickelin, Amit Singer, and Yuguan Wang. Subspace method of moments for ab initio 3-d single-particle cryo-em reconstruction. *arXiv preprint arXiv:2410.06889*, 2024.

[99] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018.

[100] Benjamin Huber, Reinhold Schneider, and Sebastian Wolf. A randomized tensor train singular value decomposition. In *Compressed Sensing and its Applications: Second International MATHEON Conference 2015*, pages 261–290. Springer, 2017.

[101] William Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and E Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and technology*, 4(2):024001, 2019.

[102] Yoonhaeng Hur, Jeremy G Hoskins, Michael Lindsey, E Miles Stoudenmire, and Yuehaw Khoo. Generative modeling via tensor train sketching. *arXiv preprint arXiv:2202.11788*, 2022.

[103] Yoonhaeng Hur, Jeremy G Hoskins, Michael Lindsey, E Miles Stoudenmire, and Yuehaw Khoo. Generative modeling via tensor train sketching. *Applied and Computational Harmonic Analysis*, 67:101575, 2023.

[104] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[105] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[106] Marina V Karsanina and Kirill M Gerke. Hierarchical optimization: Fast and robust multiscale stochastic reconstructions with rescaled correlation functions. *Physical review letters*, 121(26):265501, 2018.

[107] Kengo Kato. On the estimation in high-dimensional additive models. *arXiv preprint arXiv:1207.5313*, 2012.

[108] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Efficient construction of tensor ring representations from sampling. *arXiv preprint arXiv:1711.00954*, 2017.

[109] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Efficient construction of tensor ring representations from sampling. *Multiscale Modeling & Simulation*, 19(3):1261–1284, 2021.

[110] Yuehaw Khoo, Mathias Oster, and Yifan Peng. Optimization-free diffusion model–a perturbation theory approach. *arXiv preprint arXiv:2505.23652*, 2025.

[111] Yuehaw Khoo, Yifan Peng, and Daren Wang. Nonparametric estimation via variance-reduced sketching. *arXiv preprint arXiv:2401.11646*, 2024.

[112] Boris N Khoromskij. Tensors-structured numerical methods in scientific computing: survey on recent advances. *Chemometrics and Intelligent Laboratory Systems*, 110(1):1–19, 2012.

[113] Boris N. Khoromskij. Tensor numerical methods for high-dimensional partial differential equations: Basic theory and initial applications. *ESAIM: Proceedings and Surveys*, 48:1–28, 2015.

[114] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

[115] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[116] Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-

supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

[117] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[118] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 2019.

[119] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[120] Tamara G Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *2008 Eighth IEEE international conference on data mining*, pages 363–372. IEEE, 2008.

[121] Daniel Kressner and André Uschmajew. On low-rank approximability of solutions to high-dimensional operator equations and eigenvalue problems. *Linear Algebra and its Applications*, 493:556–572, 2016.

[122] Daniel Kressner, Bart Vandereycken, and Rik Voorhaar. Streaming tensor train approximation. *SIAM Journal on Scientific Computing*, 45(5):A2610–A2631, 2023.

[123] Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions. *Linear Algebra and its Applications*, 18(2):95–138, 1977.

[124] Joseph B Kruskal. Rank, decomposition, and uniqueness for 3-way and n-way arrays. *Multiway Data Analysis*, pages 7–18, 1989.

[125] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the nyström method. *The Journal of Machine Learning Research*, 13(1):981–1006, 2012.

[126] Andrew B Lawson. Bayesian point event modeling in spatial and environmental epidemiology. *Statistical Methods in Medical Research*, 21(5):509–529, 2012.

[127] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu-Jie Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 2006.

[128] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *Advances in Neural Information Processing Systems*, 35:22870–22882, 2022.

[129] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence of score-based generative modeling for general data distributions. In *International Conference on Algorithmic Learning Theory*, pages 946–985. PMLR, 2023.

[130] Michael D Lee. How cognitive modeling can benefit from hierarchical bayesian models. *Journal of Mathematical Psychology*, 55(1):1–7, 2011.

[131] Lexin Li and Xin Zhang. Parsimonious tensor response regression. *Journal of the American Statistical Association*, 112(519):1131–1146, 2017.

[132] Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 10(3):520–545, 2018.

[133] Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007.

[134] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[135] Han Liu, John Lafferty, and Larry Wasserman. Sparse nonparametric density estima-

tion in high dimensions using the rodeo. In *Artificial Intelligence and Statistics*, pages 283–290. PMLR, 2007.

[136] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2012.

[137] Linxi Liu, Dangna Li, and Wing Hung Wong. Convergence rates of a class of multivariate density estimation methods based on adaptive partitioning. *Journal of machine learning research*, 24(50):1–64, 2023.

[138] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

[139] Qiao Liu, Jiaze Xu, Rui Jiang, and Wing Hung Wong. Density estimation using deep generative neural networks. *Proceedings of the National Academy of Sciences*, 118(15):e2101344118, 2021.

[140] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[141] Eric F. Lock. Tensor-on-tensor regression. *Journal of Computational and Graphical Statistics*, 27(3):638–647, 2018.

[142] Wenqi Lu, Zhongyi Zhu, and Heng Lian. High-dimensional quantile tensor regression. *Journal of Machine Learning Research*, 21(250):1–31, 2020.

[143] Christian Lubich and Ivan V Oseledets. Time integration of tensor trains. *SIAM Journal on Numerical Analysis*, 53(2):917–941, 2015.

[144] Gábor Lugosi and Andrew Nobel. Consistency of data-driven histogram methods for density estimation and classification. *The Annals of Statistics*, 24(2):687–706, 1996.

[145] Zhiyuan Ma, Yuzhu Zhang, Guoli Jia, Liangliang Zhao, Yichao Ma, Mingjie Ma, Gaofeng Liu, Kaiyan Zhang, Ning Ding, Jianjun Li, et al. Efficient diffusion models: A comprehensive survey from principles to practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[146] YP Mack and Murray Rosenblatt. Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis*, 9(1):1–15, 1979.

[147] Michael W Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[148] Michael W Mahoney et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

[149] Per-Gunnar Martinsson. Randomized methods for matrix computations. *The Mathematics of Data*, 25(4):187–231, 2019.

[150] Per-Gunnar Martinsson and Joel A Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020.

[151] Lukas Meier, Sara van de Geer, and Peter Bühlmann. High-dimensional additive modeling. *The Annals of Statistics*, 37(6B):3779–3821, 2009.

[152] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[153] Zhaonan Meng, Yuehaw Khoo, Jiajia Li, and E Miles Stoudenmire. Recursive sketched interpolation: Efficient hadamard products of tensor trains. *arXiv preprint arXiv:2602.17974*, 2026.

[154] Rachel Minster, Arvind K Saibaba, and Misha E Kilmer. Randomized algorithms for low-rank tensor decompositions in the tucker format. *SIAM Journal on Mathematics of Data Science*, 2(1):189–215, 2020.

[155] Yuji Nakatsukasa. Fast and stable randomized low-rank matrix approximation. *arXiv preprint arXiv:2009.11392*, 2020.

[156] Yuji Nakatsukasa and Joel A Tropp. Fast & accurate randomized algorithms for linear systems and eigenvalue problems. *arXiv preprint arXiv:2111.00113*, 2021.

[157] Jiquan Ngiam, Zhenghao Chen, Sandeep Bhaskar, Pang Wei Koh, and Andrew Y Ng. Learning deep energy models. In *International Conference on Machine Learning*, 2011.

[158] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *International Conference on Machine Learning*, 2021.

[159] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 442–450, 2015.

[160] Georgii S Novikov, Maxim E Panov, and Ivan V Oseledets. Tensor-train density estimation. In *Uncertainty in artificial intelligence*, pages 1321–1331. PMLR, 2021.

[161] Georgii S Novikov, Maxim E Panov, and Ivan V Oseledets. Tensor-train density estimation. In *Uncertainty in Artificial Intelligence*, pages 1321–1331. PMLR, 2021.

[162] Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pages 26517–26582. PMLR, 2023.

[163] Román Orús. Advances on tensor network theory: symmetries, fermions, entanglement, and holography. *The European Physical Journal B*, 87:1–18, 2014.

[164] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[165] Ivan Oseledets. Dmrg approach to fast linear algebra in the tt-format. *Computational Methods in Applied Mathematics*, 11(3):382–393, 2011.

[166] Ivan Oseledets and Eugene Tyrtyshnikov. Tt-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.

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

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

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

[170] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[171] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[172] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[173] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[174] Grigorios A Pavliotis. Stochastic processes and applications. *Texts in applied mathematics*, 60, 2014.

[175] Yifan Peng, Yian Chen, E Miles Stoudenmire, and Yuehaw Khoo. Generative modeling via hierarchical tensor sketching. *arXiv preprint arXiv:2304.05305*, 2023.

[176] Yifan Peng, Siyao Yang, Yuehaw Khoo, and Daren Wang. Tensor density estimator by convolution-deconvolution. *arXiv preprint arXiv:2412.18964*, 2024.

[177] David Perez-Garcia, Frank Verstraete, Michael M Wolf, and J Ignacio Cirac. Matrix product state representations. *arXiv preprint quant-ph/0608197*, 2006.

[178] David Perez-Garcia, Frank Verstraete, Michael M Wolf, and J Ignacio Cirac. Matrix product state representations. *arXiv preprint quant-ph/0608197*, 2006.

[179] Zhen Qin, Alexander Lidiak, Zhexuan Gong, Gongguo Tang, Michael B Wakin, and Zhihui Zhu. Error analysis of tensor-train cross approximation. *arXiv preprint arXiv:2207.04327*, 2022.

[180] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[181] Garvesh Raskutti and Michael W Mahoney. A statistical perspective on randomized sketching for ordinary least-squares. *The Journal of Machine Learning Research*, 17(1):7508–7538, 2016.

[182] Pradeep Ravikumar, Han Liu, John Lafferty, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B*, 71(5):1009–1030, 2009.

[183] Yinuo Ren, Hongli Zhao, Yuehaw Khoo, and Lexing Ying. High-dimensional density estimation with tensorizing flow. *Research in the Mathematical Sciences*, 10(3):30, 2023.

[184] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[185] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[186] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning*, 2014.

[187] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pages 832–837, 1956.

[188] Arvind K Saibaba and Peter K Kitanidis. Efficient methods for large-scale linear inversion using a geostatistical approach. *Water Resources Research*, 48(5), 2012.

[189] Berkant Savas and Lars Eldén. Handwritten digit classification using higher order singular value decomposition. *Pattern recognition*, 40(3):993–1003, 2007.

[190] Dmitry Savostyanov and Ivan Oseledets. Fast adaptive interpolation of multi-dimensional arrays in tensor train format. In *The 2011 International Workshop on Multidimensional (nD) Systems*, pages 1–8. IEEE, 2011.

[191] Dmitry V Savostyanov. Quasioptimality of maximum-volume cross interpolation of tensors. *Linear Algebra and its Applications*, 458:217–244, 2014.

[192] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

[193] David W Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[194] David W Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[195] David W Scott. Averaged shifted histograms: effective nonparametric density estimators in several dimensions. *The Annals of Statistics*, pages 1024–1040, 1985.

[196] Richik Sengupta, Soumik Adhikary, Ivan Oseledets, and Jacob Biamonte. Tensor networks in machine learning. *European Mathematical Society Magazine*, (126):4–12, 2022.

[197] Tianyi Shi, Maximilian Ruth, and Alex Townsend. Parallel algorithms for computing the tensor-train decomposition. *arXiv preprint arXiv:2111.10448*, 2021.

[198] Tianyi Shi, Maximilian Ruth, and Alex Townsend. Parallel algorithms for computing the tensor-train decomposition. *SIAM Journal on Scientific Computing*, 45(3):C101–C130, 2023.

[199] Sungho Shin, Philip Hart, Thomas Jahns, and Victor M Zavala. A hierarchical optimization architecture for large-scale power networks. *IEEE Transactions on Control of Network Systems*, 6(3):1004–1014, 2019.

[200] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[201] Qingquan Song, Hancheng Ge, James Caverlee, and Xia Hu. Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(1):1–48, 2019.

[202] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

[203] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[204] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[205] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[206] Yang Song, Jascha Sohl-Dickstein, Durk P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[207] Michael Steinlechner. Riemannian optimization for high-dimensional tensor completion. *SIAM Journal on Scientific Computing*, 38(5):S461–S484, 2016.

[208] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. *Advances in neural information processing systems*, 29, 2016.

[209] Yiming Sun, Yang Guo, Charlene Luo, Joel Tropp, and Madeleine Udell. Low-rank tucker approximation of a tensor from streaming data. *SIAM Journal on Mathematics of Data Science*, 2(4):1123–1150, 2020.

[210] Xun Tang, Yoonhaeng Hur, Yuehaw Khoo, and Lexing Ying. Generative modeling via tree tensor network states. *arXiv preprint arXiv:2209.01341*, 2022.

[211] Xun Tang and Lexing Ying. Solving high-dimensional fokker-planck equation with functional hierarchical tensor. *arXiv preprint arXiv:2312.07455*, 2023.

[212] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

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

[214] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. *Advances in Neural Information Processing Systems*, 30, 2017.

[215] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1454–1485, 2017.

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

[217] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220, 2016.

[218] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220, 2016.

[219] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

[220] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12(1), 2016.

[221] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European conference on computer vision*, pages 447–460. Springer, 2002.

[222] Frank Verstraete and J Ignacio Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*, 2004.

[223] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

[224] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[225] Haiyong Wang. How much faster does the best polynomial approximation converge than legendre projection? *Numerische Mathematik*, 147(2):481–503, 2021.

[226] Shusen Wang, Alex Gittens, and Michael W Mahoney. Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. In *International Conference on Machine Learning*, pages 3608–3616. PMLR, 2017.

[227] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5697–5705, 2017.

[228] Wenqi Wang, Guoxu Zhou, Qibin Zhao, and Andrzej Cichocki. Efficient low-rank tensor ring completion. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):4638–4652, 2018.

[229] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[230] Per-rAke Wedin. Perturbation theory for pseudo-inverses. *BIT Numerical Mathematics*, 13:217–232, 1973.

[231] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.

[232] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.

[233] Steven R White. Density-matrix algorithms for quantum renormalization groups. *Physical review b*, 48(14):10345, 1993.

[234] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

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

[236] Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. 2008.

[237] Yuan Xu. Approximation by polynomials in sobolev spaces with jacobi weight. *Journal of Fourier Analysis and Applications*, 24:1438–1459, 2018.

[238] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[239] Yun Yang, Mert Pilanci, and Martin J Wainwright. Randomized sketches for kernels: Fast and optimal nonparametric regression. *Annals of Statistics*, pages 991–1023, 2017.

[240] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. *Advances in Neural Information Processing Systems*, 37:22370–22417, 2024.

[241] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023.

[242] Ming Yuan and Ding-Xuan Zhou. Minimax optimal rates of estimation in high dimensional additive models. *Annals of Statistics*, 44(6):2564–2593, 2016.

[243] Adriano Z Zambom and Ronaldo Dias. A review of kernel density estimation with applications to econometrics. *International Econometric Review*, 5(1):20–42, 2013.

[244] Qibin Zhao, Guoxu Zhou, Shuo Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

[245] Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

[246] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.