Smooth tensor estimation with unknown permutations

Abstract

We consider the problem of structured tensor denoising in the presence of unknown permutations. Such data problems arise commonly in recommendation system, neuroimaging, community detection, and multiway comparison applications. Here, we develop a general family of smooth tensor models up to arbitrary index permutations; the model incorporates the popular tensor block models and Lipschitz hypergraphon models as special cases. We show that a constrained least-squares estimator in the block-wise polynomial family achieves the minimax error bound. A phase transition phenomenon is revealed with respect to the smoothness threshold needed for optimal recovery. In particular, we find that a polynomial of degree up to $(m-2)(m+1)/2$ is sufficient for accurate recovery of order-m tensors, whereas higher degree exhibits no further benefits. This phenomenon reveals the intrinsic distinction for smooth tensor estimation problems with and without unknown permutations. Furthermore, we provide an efficient polynomial-time Borda count algorithm that provably achieves optimal rate under monotonicity assumptions. The efficacy of our procedure is demonstrated through both simulations and Chicago crime data analysis.

Keywords: Tensor estimation, latent permutation, diverging dimensionality, phase transition, statistical-computational efficiency.

1 Introduction

Higher-order tensor datasets are rising ubiquitously in modern data science applications, for instance, recommendation systems [4], social networks [5], and genomics [32]. Tensor provides effective representation of data structure that classical vector- and matrix-based methods fail to capture. For example, the music recommendation system [2] records ratings of songs from users on various contexts. This three-way tensor of user $\times$ song $\times$ context allows us to investigate interactions of users and songs in a context-specific manner.

Tensor estimation problem cannot be solved without imposing structures. An appropriate reordering of tensor entries often provides effective representation of the hidden structure. In the music recommendation example, suppose that we have certain criteria available (such as, similarities of music genres, ages of users, and importance of contexts) to reorder the songs, users, and contexts. Then, the sorted tensor will exhibit smooth structure, because entries from similar groups tend to have similar values.

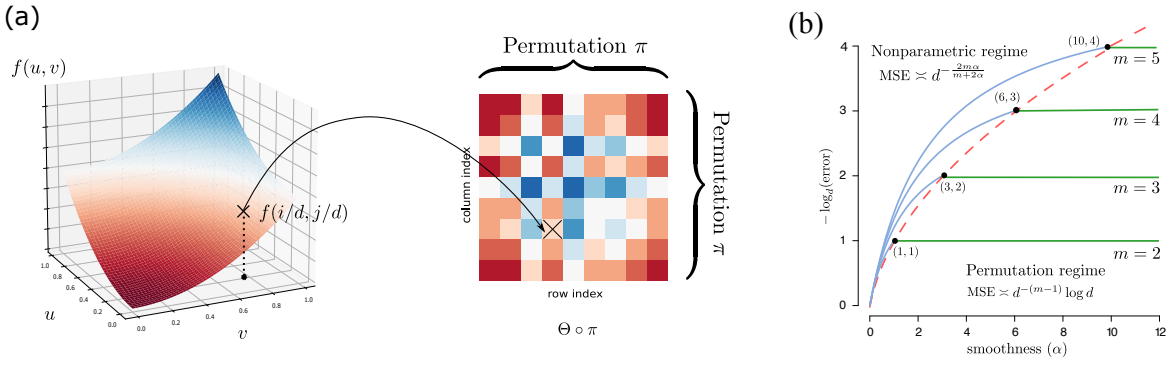


Figure 1: (a): Illustration of order-m d-dimensional permuted smooth tensor models with $m = 2$. (b): Phase transition of mean squared error (MSE) (on $-\log_d$ scale) as a function of smoothness α and tensor order m. Bold dots correspond to the critical smoothness level above which higher smoothness exhibits no further benefits to tensor estimation.

In this article, we develop a *permuted* smooth tensor model based on the aforementioned

motivation. We study a class of structured tensors, called *permuted smooth tensor model*, of the following form:

$$\mathcal{Y} = \Theta \circ \pi + \text{noise}, \quad \text{where} \quad \Theta(i_1, \ldots, i_m) = f\left(\frac{i_1}{d}, \ldots, \frac{i_m}{d}\right), \tag{1}$$

where $\pi\colon [d] \to [d]$ is an *unknown* latent permutation, Θ is an *unknown* order-m d-dimensional signal tensor, and f is an *unknown* multivariate function with certain notion of smoothness, and $\Theta \circ \pi$ denotes the permuted tensor after reordering the indices along each of the m modes. Figure 1(a) shows an example of this generative model for the matrix case $m = 2$. Our primary goal is to estimate the permuted smooth signal tensor $\Theta \circ \pi$ from the noisy tensor observation $\mathcal{Y}$ of arbitrary order m.

1.1 Our contributions

We develops a suite of statistical theory, efficient algorithms, and related applications for permuted smooth tensor models (1). Our contributions are summarized below.

	Pananjady and Samworth [26]	Balasubramanian [1]	Li et al. [22]	**Ours** (MLE)	**Ours** (Borda count)
Model structure	monotonic	Lipschitz	Lipschitz	α-smooth	α-smooth & monotonic
Fixed grid design	$\surd$	$\times$	$\times$	$\times$	$\surd$
Error rate for order-m tensor	d^{-1}	$d^{-\frac{2m}{m+2}}$	$d^{-\lfloor m/3 \rfloor}$	$d^{-\frac{2m\alpha}{m+2\alpha}} \vee d^{-(m-1)}$	$d^{-\frac{2m\alpha}{m+2\alpha}} \vee d^{-(m-1)}$
(e.g., when $(m, \alpha) = (3, 1)$)		$(d^{-6/5})$	(d^{-1})	$(d^{-6/5})$	$(d^{-6/5})$
Minimax optimality	$\surd$	$\times$	$\times$	$\surd$	$-*$
Polynomial algorithm	$\surd$	$\times$	$\surd$	$\times$	$\surd$

Table 1: Comparison of our results with previous work. For simplicity, we omit the log term in the rate. *The optimality is achieved under extra Lipchitz monotonicity conditions.

First, we develop a general permuted α-smooth tensor model of arbitrary smoothness level $\alpha > 0$. We establish the statistically optimal error rate and its dependence on model complexity. Specifically, we express the optimal rate as a function of tensor order m, tensor

dimension d, and the smoothness level α, given by

$$\text{Rate}(d) := d^{-\frac{2m\alpha}{m+2\alpha}} \vee d^{-(m-1)} \log d. \tag{2}$$

Table 1 summarizes the comparison of our work with previous results. Our framework substantially generalizes earlier works which focus on only matrices with $m = 2$ [16, 19] or Lipschitz function with $\alpha = 1$ [1, 22]. The generalization enables us to obtain results previously impossible: i) As tensor order m increases, we demonstrate the failure of pervious clustering-based algorithms [1, 16], and we develop a new block-wise polynomial algorithm for tensors of order $m \geq 3$; ii) As smoothness α increases, we demonstrate that the error rate converges to a fast rate $\mathcal{O}(d^{-(m-1)})$, thereby disproving the conjectured lower bound $\mathcal{O}(d^{-2m/(m+2)})$ posed by earlier work [1]. The results showcase the accuracy gain of our new approach, as well as the intrinsic distinction between matrices and higher-order tensors.

Second, we discover a phase transition phenomenon with respect to the smoothness needed for optimal recovery in model (1). Figure 1(b) plots the dependence of estimation error in terms of smoothness level α for tensors of order m. We characterize two distinct error behaviors determined by a critical smoothness threshold; see Theorems 1-2 in Section 4. Specifically, the accuracy improves with α in the regime $\alpha \leq m(m-1)/2$, whereas the accuracy becomes a constant of α in the regime $\alpha > m(m-1)/2$. The results imply a polynomial of degree $(m-2)(m+1)/2 = [m(m-1)/2 - 1]$ is sufficient for accurate recovery of order-m tensors of arbitrary smoothness in model (1), whereas higher degree brings no further benefits. The phenomenon is distinctive from matrix problems [19, 16] and classical *non-permuted* smooth function estimation [31], thereby highlighting the fundamental challenges in our new setting. These statistical contributions, to our best knowledge, are new to the literature of general permuted smooth tensor problems.

Third, we propose two estimation algorithms with accuracy guarantees: the least-squares

estimation and Borda count estimation. The least-squares estimation, although being computationally hard, reveals the fundamental model complexity in the problem. The result serves as the benchmark and a useful guide to the algorithm design. Furthermore, we develop an efficient polynomial-time Borda count algorithm that provably achieves a minimax optimal rate under an extra Lipschitz monotonic assumption. The algorithm handles a broad range of data types, including continuous and binary observations.

Lastly, we illustrate the efficacy of our method through both simulations and data applications. A range of practical settings are investigated in simulations, and we show the outperformance of our method compared to alternative approaches. Application to Chicago crime data is presented to showcase the usefulness of our method. We identify the key global pattern and pinpoint local smooth structure in the denoised tensor. Our method will help practitioners efficiently analyze tensor datasets in various areas. Toward this end, the package and all data used are released at CRAN.

1.2 Related work

Our work is closely related to but also clearly distinctive from several lines of existing research. We review related literature for comparison.

Structure learning with latent permutations. The estimation problem of (1) falls into the general category of structured learning with *latent permutation*. Models involving latent permutations have recently received a surge of interest, include graphons [7, 19], stochastic transitivity models [8, 28], statistical seriation [14], and graph matching [11]. These methods, however, are developed for matrices; the tensor counterparts are far less well understood. Table 1 summarizes the most related works to ours. Pananjady and Samworth [26] studied the permuted tensor estimation under isotonic constraints. We find that our smooth model

results in a much faster rate $\mathcal{O}(d^{-(m-1)})$ than the rate $\mathcal{O}(d^{-1})$ for isotonic models. The works [1, 22] studied similar smooth models as ours, but we gain substantial improvement in both statistics and computations. Balasubramanian [1] developed a (non-polynomial-time) clustering-based algorithm with a rate $\mathcal{O}(d^{-2m/(m+2)})$. Li et al. [22] developed a (polynomial-time) nearest neighbor estimation with a rate $\mathcal{O}(d^{-\lfloor m/3 \rfloor})$. Neither approach investigates the minimax optimality. By contrast, we develop a polynomial-time algorithm with a fast rate $\mathcal{O}(d^{-(m-1)})$ under mild conditions. The optimality of our estimator is safeguarded by matching a minimax lower bound.

Low-rank tensor models. There is a huge literature on structured tensor estimation under low-rank models, including CP models [20], Tucker models [37], and block models [34]. These models belong to parametric approaches, because they aim to explain the data with a finite number of parameters (i.e., decomposed factors). Our permuted smooth tensor model utilizes a different measure of model complexity than the usual low-rankness. We use *infinite* number of parameters (i.e., smooth functions) to allow growing model complexity. In this sense, our method belongs to nonparametric approaches. We emphasize that our permuted smooth tensor model does not necessarily include low-rank model. Compared to low-rank models, we utilize a different measure of *model complexity*. When the underlying signal is precisely low-rank, then low-rank methods are preferred. However, if the underlying signal is high rank but has certain shape structure, then our nonparametric approach better captures the intrinsic model complexity.

Nonparametric regression. Our model is also related to nonparametric regression [31]. One may view the problem (1) as a nonparametric regression, where the goal is to learn the function f based on scalar response $\mathcal{Y}(i_1, \ldots, i_m)$ and design points $(\pi(i_1), \ldots, \pi(i_m))$ in $\mathbb{R}^m$;

see Figure 1(a). However, the *unknown* permutation π significantly influences the statistical and computational hardness of the problem. This latent π leads to a phase transition behavior in the estimation error; see Figure 1(b) and Section 4. We reveal two components of error for the problem, one for nonparametric error and the other for permutation error. The impact of unknown permutation hinges on tensor order and smoothness in an intriguing way, as shown in (2). This is clearly contrary to classical nonparametric regression.

Graphon and hypergraphon. Our work is also connected to graphons and hypergraphons. Graphon is a measurable function representing the limit of a sequence of exchangeable random graphs (matrices) [19, 16, 7]. Similarly, hypergraphon [38, 23] is introduced as a limiting function of m-uniform hypergraphs, i.e., a generalization of graphs in which edges can join m vertices with $m \geq 3$. While both our model (1) and hypergraphon focus on function representations, there are two remarkable differences. First, unlike the matrix case where graphon is represented as bivariate functions [23], hypergraphons for m-uniform hypergraphs should be represented as $(2^m - 2)$-multivariate functions; see Zhao [38, Section 1.2]. Our framework (1) represents the function using m coordinates only, and in this sense, the model shares the common ground as *simple hypergraphons* [1]. We compare our method to earlier work in theory (Table 1 and Sections 4-5) and in numerical studies (Section 6). Second, unlike typical simple hypergraphons where the design points are random, our generative model uses deterministic design points. The comparison of two approaches will be discussed in Sections 2 and 4.

1.3 Notation and organization

Let $\mathbb{N}, \mathbb{N}_+$ denote the set of non-negative integers and positive integers, respectively. We use $[d] = \{1, \ldots, d\}$ to denote the d-set for $d \in \mathbb{N}_+$. For a set S, $|S|$ denotes its cardinality and

$\mathbb{1}_S$ denotes the indicator function. For two positive sequences $\{a_d\}, \{b_d\}$, we denote $a_d \lesssim b_d$ if $\lim_{d\to\infty} a_d/b_d \leq c$ for some constant $c > 0$, and $a_d \asymp b_d$ if $c_1 \leq \lim_{d\to\infty} a_d/b_d \leq c_2$ for some constants $c_1, c_2 > 0$. Given a number $a \in \mathbb{R}$, the function $\lfloor a \rfloor$ is the largest integer strictly smaller than a and the ceiling function $\lceil a \rceil$ is the smallest integer no less than a. We use $\mathcal{O}(\cdot)$ to denote big-O notation hiding logarithmic factors, and $\circ$ the function composition.

Let $\Theta \in \mathbb{R}^{d\times\cdots\times d}$ be an order-m d-dimensional tensor, $\pi\colon [d] \to [d]$ be an index permutation, and $\Theta(i_1, \ldots, i_m)$ the tensor entry indexed by $(i_1, \ldots, i_m)$. We sometimes also use shorthand notation $\Theta(\omega)$ for tensor entries with indices $\omega = (i_1, \ldots, i_m)$. We call a tensor *a binary-valued tensor* if its entries take value on $\{0, 1\}$-labels, and *a continues-valued tensor* if its entries take values on a continuous scale. We define the Frobenius norm $\|\Theta\|_F^2 = \sum_{\omega\in[d]^m} |\Theta(\omega)|^2$ for a tensor Θ, and the ∞-norm $\|\boldsymbol{x}\|_\infty = \max_{i\in[d]} |x_i|$ for a vector $\boldsymbol{x} = (x_1, \ldots, x_d)^T$. We use $\Pi(d, d) = \{\pi\colon [d] \to [d]\}$ to denote all permutations on $[d]$, while $\Pi(d, k) = \{\pi\colon [d] \to [k]\}$ the collection of all onto mappings from $[d]$ to $[k]$. Given $\pi \in \Pi(d, k)$ and $\Theta \in \mathbb{R}^{k\times\cdots\times k}$, we use $\Theta \circ \pi$ to denote the d-dimensional tensor such that $(\Theta \circ \pi)(i_1, \ldots, i_m) = \Theta(\pi(i_1), \ldots, \pi(i_m))$ for all $(i_1, \ldots, i_m) \in [d]^m$. An event A is said to occur *with high probability* if $\mathbb{P}(A)$ tends to 1 as the tensor dimension $d \to \infty$.

The rest of the paper is organized as follows. Section 2 presents the permuted smooth tensor model and its connection to smooth function representation. In Section 3, we establish the approximation error based on block-wise polynomial approximation. Then, we develop two estimation algorithms with accuracy guarantees: the least-squares estimation and Borda count estimation. Section 4 presents a statistically optimal but computationally challenging least-squares estimator. Section 5 presents a polynomial-time Borda count algorithm with the same minimax optimal rate under an extra Lipschitz monotonic assumption. Simulations and data analyses are presented in Section 6. We conclude the paper with a discussion in

Section 7. All proofs and extensions are deferred to Appendix.

2 Smooth tensor model with unknown permutation

Suppose we observe an order-m d-dimensional data tensor from the following model,

$$\mathcal{Y} = \Theta \circ \pi + \mathcal{E}, \tag{3}$$

where $\pi\colon [d] \to [d]$ is an unknown latent permutation, $\Theta \in \mathbb{R}^{d\times\cdots\times d}$ is an unknown signal tensor under certain smoothness (to be specified in next paragraph), and $\mathcal{E}$ is a noise tensor consisting of zero-mean, independent sub-Gaussian entries with variance bounded by σ^2. The general model allows continuous- and binary-valued tensors. For instance, in binary tensor problems, the entries in $\mathcal{Y}$ are $\{0,1\}$-labels from Bernoulli distribution, and the entrywise variance of $\mathcal{E}$ depends on the mean. For ease of presentation, we assume $\sigma = 1$ throughout the paper. We call (3) the Gaussian model if the $\mathcal{E}$ consists of i.i.d. $\mathcal{N}(0,1)$ entries, and call (3) the sub-Gaussian model if $\mathcal{E}$ consists of independently (but not necessarily identically) distributed sub-Gaussian entries.

We now describe the smooth model on the signal Θ. Suppose that there exists a multivariate function $f\colon [0,1]^m \to \mathbb{R}$ underlying the signal tensor, such that

$$\Theta(i_1,\ldots,i_m) = f\left(\frac{i_1}{d},\ldots,\frac{i_m}{d}\right), \quad \text{for all } (i_1,\ldots,i_m) \in [d]^m. \tag{4}$$

For a multi-index $\kappa = (\kappa_1,\ldots,\kappa_m) \in \mathbb{N}^m$ and a vector $\boldsymbol{x} = (x_1,\ldots,x_m)^T$, we denote $|\kappa| = \sum_{i\in[m]}\kappa_i$, $\kappa! = \prod_{i\in[m]}\kappa_i!$, $\boldsymbol{x}^\kappa = \prod_{i\in[m]} x_i^{\kappa_i}$, and the derivative operator $\nabla_\kappa = \frac{\partial^{|\kappa|}}{\partial x_1^{\kappa_1}\cdots\partial x_m^{\kappa_m}}$. Assume the generative function f in (4) is in the α-Hölder smooth family [35, 31].

Definition 1 (α-Hölder smooth)**.** Let $\alpha > 0$ and $L > 0$ be two positive constants. A function $f\colon [0,1]^m \to \mathbb{R}$ is called α-Hölder smooth, denoted as $f \in \mathcal{F}(\alpha, L)$, if for every

$\boldsymbol{x}, \boldsymbol{x}_0 \in [0,1]^m$, we have

$$\sum_{\kappa:|\kappa|=\lfloor\alpha\rfloor} \frac{1}{\kappa!}|\nabla_\kappa f(\boldsymbol{x}) - \nabla_\kappa f(\boldsymbol{x}_0)| \leq \begin{cases} L\|\boldsymbol{x}-\boldsymbol{x}_0\|_\infty, & \text{if } |\kappa| < \lfloor\alpha\rfloor, \\ L\|\boldsymbol{x}-\boldsymbol{x}_0\|_\infty^{\alpha-\lfloor\alpha\rfloor}, & \text{if } |\kappa| = \lfloor\alpha\rfloor. \end{cases}$$

The Hölder smooth function class is one of the most popular function classes considered in the nonparametric regression literature [19, 16]. In addition to the function class $\mathcal{F}(\alpha, L)$, we also define the smooth tensor class based on discretization (4),

$$\mathcal{P}(\alpha, L) = \left\{\Theta \in \mathbb{R}^{d\times\cdots\times d} \colon \Theta \text{ is generated from (4) and } f \in \mathcal{F}(\alpha, L)\right\}.$$

Combining (3) and (4) yields our proposed *permuted smooth tensor model*. The unknown parameters are the smooth tensor $\Theta \in \mathcal{P}(\alpha, L)$ and latent permutation $\pi \in \Pi(d, d)$. The generative model is visualized in Figure 1(a) for the case $m = 2$ (matrices). We give two examples to show the applicability of our permuted smooth tensor model.

Example 1 (Four-player game tensors)**.** Consider the tournament of a four-player board game. Suppose there are in total d players, among which all combinations of four have played with each other. The tournament results are summarized as an order-4 (non-symmetric) tensor, with entries encoding the winner out of the four. Our model is then given by

$$\mathbb{E}\mathcal{Y}(i_1, \ldots, i_4) = \mathbb{P}(\text{player } i_1 \text{ wins over } (i_2, i_3, i_4)) = f\left(\frac{\pi(i_1)}{d}, \cdots, \frac{\pi(i_4)}{d}\right).$$

We interpret the permutation π as the unknown ranking among d players, and the function f as the unknown four-player interaction. Players with similar ranking may have similar performance reflected by the smoothness of f. For example, a variant of popular Plackett-Luce model [10] considers the parametric form $f(x_1, x_2, x_3, x_4) = \exp(\beta x_1)/\sum_{i=1}^4 \exp(\beta x_i)$. By contrast, our model leaves the form of f unspecified, and we learn the function from the data in a nonparametric approach.

Example 2 (Co-authorship networks)**.** Consider a co-authorship network consisting of d nodes (authors) in total. We say there exists a hyperedge of size m between nodes $(i_1, \ldots, i_m)$ if the authors $i_1, \ldots, i_m$ have co-authored at least one paper. The resulting m-uniform hypergraph is represented as an order-m (symmetric) adjacency tensor. Our model is then expressed as

$$\mathbb{E}\mathcal{Y}(i_1, \ldots, i_m) = \mathbb{P}(\text{authors } i_1, \ldots, i_m \text{ co-authored}) = f\left(\frac{\pi(i_1)}{d}, \cdots, \frac{\pi(i_m)}{d}\right).$$

We interpret the permutation π as the affinity measures of authors, and the function f represents the m-way interaction among authors. The parametric model [33] imposes logistic function $f(x_1, \ldots, x_m) = (1+\exp(-\beta x_1 x_2 \cdots x_m))^{-1}$. By contrast, our nonparametric model allows unknown f and learns the function directly from data.

Our model (4) assumes equally-spaced grid design $\{1/d, 2/d, \ldots, d/d\}$ from the generative function f. One can also extend the model to non-equally-spaced designs. Specifically, suppose that the signal tensor is generated from f based on

$$\Theta(i_1, \ldots, i_m) = f(x_{i_1}, \ldots, x_{i_m}), \quad \text{for all} \quad (i_1, \ldots, i_m) \in [d]^m. \tag{5}$$

where the points $\{x_i\}_{i=1}^d$ can be modeled as either fixed latent variables or i.i.d. random variables from a probability distribution supported on $[0, 1]$. We refer to (5) as the relaxed smooth model; similar model has been developed in the literature of graphons and hypergraphons [7, 16, 19, 1]. Our paper will focus on the grid design (4), but we will also discuss the extension to (5) whenever possible as remarks of theorems.

We restrict ourself to tensor models with equal dimension and same permutation along m modes. The results for non-symmetric tensors with m distinct permutations are similar but require extra notations; we assess this general case in Appendix A.2.

3 Block-wise tensor estimation

Our general strategy for estimating the permuted smooth tensor is based on the block-wise tensor approximation. In this section, we first introduce the tensor block model [34, 17]. Then, we extend this model to block-wise polynomial approximation.

3.1 Tensor block model

Tensor block models describe a checkerboard pattern in a signal tensor. The block model provides a meta structure to many popular models including the low-rankness [15] and isotonic tensors [26]. Here, we use tensor block models as a building block for estimating permuted smooth models.

Specifically, suppose that there are k clusters among d entities, and the cluster assignment is represented by a clustering function $z\colon [d] \to [k]$. Then, the tensor block model assumes that the entries of signal tensor $\Theta \in \mathbb{R}^{d\times\cdots\times d}$ take values from a core tensor $\mathcal{S} \in \mathbb{R}^{k\times\cdots\times k}$ according to the clustering function z; that is,

$$\Theta(i_1, \ldots, i_m) = \mathcal{S}(z(i_1), \ldots, z(i_m)), \quad \text{for all } (i_1, \ldots, i_m) \in [d]^m. \tag{6}$$

Here, the core tensor $\mathcal{S}$ collects the entry values of m-way blocks; the core tensor $\mathcal{S}$ and clustering function $z \in \Pi(d, k)$ are parameters of interest. A tensor Θ satisfying (6) is called a block-k tensor. Tensor block models allow various data types, as shown below.

Example 3 (Gaussian tensor block model)**.** Let $\mathcal{Y}$ be a continuous-valued tensor. The Gaussian tensor block model draws independent normal entries according to $\mathcal{Y}(i_1, \ldots, i_m) \overset{\text{ind}}{\sim} N(\mathcal{S}(z(i_1), \ldots, z(i_m)), \sigma^2)$. The mean model belongs to (6), and noises entries are i.i.d. $N(0, \sigma^2)$. The Gaussian tensor block model has served as the statistical foundation for many tensor clustering algorithms [34, 17].

Tensor block models have shown great success in discovering hidden group structures for many applications [34, 17]. Despite the popularity, the constant block assumption is insufficient to capture delicate structure when the signal tensor is complicated. This parametric model aims to explain data with a finite number of blocks; such an approach is useful when the sample outsizes the parameters. Our nonparametric model (4), by contrast, uses infinite number of parameters (i.e., smooth functions) to allow growing model complexity. We change the goal of tensor block model from clustering to approximating the generative function f in (4). In our setting, the number of blocks k should be interpreted as a resolution parameter (i.e., a bandwidth) of the approximation, similar to the notion of number of bins in histogram and polynomial regression [35].

3.2 Block-wise polynomial approximation

The block tensor (6) can be viewed as a discrete version of piece-wise *constant* function. This connection motivates us to use block-wise *polynomial* tensors to approximate α-Hölder functions. Now we extend (6) to block-wise polynomial models.

We introduce some additional notations. For a given block number k, we use $z \in \Pi(d, k)$ to denote the canonical clustering function that partitions $[d]$ into k equally-sized clusters such that $z(i) = \lceil ki/d \rceil$, for all $i \in [d]$. The collection of inverse images $\{z^{-1}(j) \colon j \in [k]\}$ is a partition of $[d]$ into k disjoint and equal-sized subsets. We use $\mathcal{E}_k$ to denote the m-way partition, i.e., a collection of k^m disjoint and equal-sized subsets in $[d]^m$, such that

$$\mathcal{E}_k = \{z^{-1}(j_1) \times \cdots \times z^{-1}(j_m) \colon (j_1, \ldots, j_m) \in [k]^m\}.$$

Let $\Delta \in \mathcal{E}_k$ denote the element in $\mathcal{E}_k$. We propose to approximate the signal tensor Θ in (4) by degree-ℓ polynomial tensors within each block $\Delta \in \mathcal{E}_k$. Specifically, let $\mathscr{B}(k, \ell)$ denote

the class of block-k degree-ℓ polynomial tensors,

$$\mathscr{B}(k,\ell) = \left\{ \mathcal{B} \in \mathbb{R}^{d\times\cdots\times d} \colon \mathcal{B}(\omega) = \sum_{\Delta\in\mathcal{E}_k} \mathrm{Poly}_{\ell,\Delta}(\omega)\mathbb{1}\{\omega\in\Delta\} \text{ for all } \omega\in[d]^m \right\}, \tag{7}$$

where $\mathrm{Poly}_{\ell,\Delta}(\cdot)$ denotes a degree-ℓ polynomial function in $\mathbb{R}^m$, with coefficients depending on block Δ; that is, a constant function $\mathrm{Poly}_{0,\Delta}(\omega) = \beta^0_\Delta$ for $\ell = 0$, a linear function $\mathrm{Poly}_{1,\Delta}(\omega) = \langle \boldsymbol{\beta}_\Delta, \omega\rangle + \beta^0_\Delta$ for $\ell = 1$, and so on so forth. Here β^0_Δ and $\boldsymbol{\beta}_\Delta$ denote unknown coefficients in polynomial function. Note that the degree-0 polynomial block tensor reduces to the constant block model (6). We generalize the constant block model to degree-ℓ polynomial block tensor (7), analogous to the generalization from k-bin histogram to k-piece-wise polynomial regression in nonparametric statistics [35].

Smoothness of the function f in (4) plays an important role in the block-wise polynomial approximation. The following lemma explains the role of smoothness in the approximation.

Proposition 1 (Block-wise polynomial tensor approximation)**.** *Suppose $\Theta \in \mathcal{P}(\alpha, L)$. Then, for every block number $k \le d$ and degree $\ell \in \mathbb{N}$, we have the approximation error*

$$\inf_{\mathcal{B}\in\mathscr{B}(k,\ell)} \frac{1}{d^m}\|\Theta - \mathcal{B}\|_F^2 \lesssim \frac{L^2}{k^{2\min(\alpha,\ell+1)}}.$$

Proposition 1 implies that we can always find a block-wise polynomial tensor close to the signal tensor generated from α-Hölder smooth function f. The approximation error decays with block number k and degree $\min(\alpha, \ell+1)$.

4 Statistical limits via least-squares estimation

We develop two estimation methods based on the block-wise polynomial approximation. We first introduce a statistically optimal but computationally inefficient least-squares estimator. The least-squares estimation serves as a statistical benchmark because of its minimax

optimality. In Section 5, we will present a polynomial-time algorithm with a provably same optimal rate under monotonicity assumptions.

We propose the least-squares estimation for model (3) by minimizing the Frobenius loss over the block-k degree-ℓ polynomial tensor family $\mathscr{B}(k,\ell)$ up to permutations,

$$(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}}) = \underset{\Theta\in\mathscr{B}(k,\ell),\pi\in\Pi(d,d)}{\arg\min} \|\mathcal{Y} - \Theta\circ\pi\|_F. \tag{8}$$

The least-squares estimator $(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}})$ depends on two tuning parameters: the number of blocks k and the polynomial degree ℓ. The optimal choice (k^*, ℓ^*) will be provided below.

Theorem 1 establishes the error bound for the least-squares estimator (8). Note that Θ and π are in general not separably identifiable; for example, when the true signal is a constant tensor, then every permutation $\pi \in \Pi(d,d)$ gives equally good fit in statistics. We assess the estimation error on the composition $\Theta\circ\pi$ to avoid this identifiability issue. For two order-m d-dimensional tensors Θ_1, Θ_2, define the mean squared error (MSE) by $\mathrm{MSE}(\Theta_1,\Theta_2) = d^{-m}\|\Theta_1 - \Theta_2\|_F^2$.

Theorem 1 (Least-squares estimation error)**.** *Let $m \geq 2$. Consider the sub-Gaussian model* (3) *with $\Theta \in \mathcal{P}(\alpha, L)$. Let $(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}})$ denote the least-squares estimator in* (8) *with a given (k,ℓ). Then, for every $k \leq d$ and degree $\ell \in \mathbb{N}$, we have*

$$\mathrm{MSE}(\hat{\Theta}^{\mathrm{LSE}}\circ\hat{\pi}^{\mathrm{LSE}},\ \Theta\circ\pi) \lesssim \underbrace{\frac{L^2}{k^{2\min(\alpha,\ell+1)}}}_{\text{approximation error}} + \underbrace{\frac{k^m(\ell+m)^\ell}{d^m}}_{\text{nonparametric error}} + \underbrace{\frac{\log d}{d^{m-1}}}_{\text{permutation error}} \tag{9}$$

with high probability. In particular, setting $\ell^* = \min(\lfloor\alpha\rfloor, (m-2)(m+1)/2)$ and $k^* = c_1 d^{m/(m+2\min(\alpha,\ell^*+1))}$ yields the optimized error rate

$$(9) \lesssim \mathrm{Rate}(d) = \begin{cases} c_2 d^{-\frac{2m\alpha}{m+2\alpha}}, & \text{when } \alpha < m(m-1)/2, \\ c_3 d^{-(m-1)}\log d, & \text{when } \alpha \geq m(m-1)/2. \end{cases} \tag{10}$$

Here, the function $\mathrm{Rate}(d)$ *is given in* (2)*, and the constants $c_1, c_2, c_3 > 0$ depend on the model configuration (m, L, α) but not on the tensor dimension d.*

We discuss the asymptotic error rate as $d \to \infty$ while treating other model configurations fixed. The final least-squares estimation rate (10) has two sources of error: the nonparametric error $d^{-\frac{2m\alpha}{m+2\alpha}}$ and the permutation error $d^{-(m-1)} \log d$. Intuitively, in the tensor data analysis problem, we can view each tensor entry as a data point, so sample size is the total number of entries, d^m. The unknown permutation results in $\log(d!) \approx d \log d$ complexity, whereas the unknown generative function results in $d^{-2m\alpha/(m+2\alpha)}$ nonparametric complexity. When the function f is smooth enough, estimating the function f becomes relatively easier compared to estimating the permutation π. This intuition coincides with the fact that the permutation error dominates the nonparametric error when $\alpha \geq m(m-1)/2$.

Remark 1 (Comparison to non-parametric regression)**.** We now compare our results with existing work in the literature. In the vector case with $m = 1$, our model reduces to the one-dimensional regression problem such that $y_i = \theta_{\pi(i)} + \epsilon_i,$ for all $i \in [d]$, where $\theta_i = f(i/d)$ and unknown $\pi \in \Pi(d, d)$. A similar analysis of Theorem 1 shows the error rate

$$\frac{1}{d} \sum_{i \in [d]} (\hat{\theta}_i^{\mathrm{LSE}} - \theta_i)^2 \lesssim \left(d^{-\frac{2\alpha}{2\alpha+1}} + \log d \right), \tag{11}$$

under the choice of $\ell^* = 0$ and $k^* \asymp d^{\frac{1}{1+2\min(\alpha,1)}}$. Notice that $d^{-2\alpha/(2\alpha+1)}$ is the classical nonparametric minimax rate for α-Hölder smooth functions [31] with *known* permuted design points $\{\pi(i)\}_{i=1}^d$. By contrast, our model involves *unknown* π, which results in the non-vanishing permutation rate $\log d$ in (11).

Remark 2 (Breaking previous limits on matrices/tensors)**.** In the matrix case with $m = 2$, Theorem 1 implies that the best rate is obtained under $\ell^* = 0$, i.e., the block-wise *constant* approximation. This result is consistent with existing literature on smooth graphons [5, 16, 19], where constant block model (see Section 3.1) is developed for estimation. In the tensor case with $m \geq 3$, earlier work [1] suggests that constant block approximation

($\ell^* = 0$) may remain minimax optima. Our Theorem 1 disproves this conjecture, and we reveal a much faster rate $d^{-(m-1)}$ compared to the conjectured lower bound $d^{-2m/(m+2)}$ [1] for sufficiently smooth tensors. We demonstrate that a polynomial up to degree $(m-2)(m+1)/2$ is sufficient (and necessary; see Theorem 2 below) for accurate estimation of order-m permuted smooth tensors. For example, permuted α-smooth tensors of order-3 require quadratic approximation ($\ell^* = 2$) with $k^* \asymp d^{1/3}$ blocks, for all $\alpha \geq 2$. The results show the clear difference between matrices and higher-order tensors.

We now show that the rate in (10) cannot be improved. The lower bound is information-theoretical and thus applies to all estimators including, but not limited to, the least-squares estimator (8) and Borda count estimator introduced in next section.

Theorem 2 (Minimax lower bound)**.** *The estimation problem based on the Gaussian model* (1) *obeys the minimax lower bound*

$$\inf_{(\hat{\Theta},\hat{\pi})} \sup_{\Theta \in \mathcal{P}(\alpha,L), \pi \in \Pi(d,d)} \mathbb{P}\left(\mathrm{MSE}(\hat{\Theta} \circ \hat{\pi},\ \Theta \circ \pi) \gtrsim \mathrm{Rate}(d)\right) \geq 0.8, \tag{12}$$

where the function $\mathrm{Rate}(d)$ *is given in* (2)*.*

The lower bound in (12) matches the upper bound in (10), demonstrating the statistical optimality of the convergence speed $\mathrm{Rate}(d) = d^{-\frac{2m\alpha}{2\alpha+m}} \vee d^{-(m-1)} \log d$. The two-component error reveals the intrinsic model complexity. In particular, the permutation error $d^{-(m-1)}$ dominates nonparametric error $d^{-2m\alpha/(m+2\alpha)}$ for sufficiently smooth tensors. This phenomenon is contrary to classical nonparametric regression.

Remark 3 (Phase transition)**.** Our Theorem 1 can be extended to relaxed smooth model (5). In the Appendix F, we show that the relaxed smooth model has the same convergence upper bound $\mathrm{Rate}(d)$, with little modification of proofs. We conclude this section by summarizing an interesting phase transition phenomenon. Figure 1(b) plots the optimal convergence speed

Rate(d). The impact of unknown permutation hinges on the tensor order and smoothness. The accuracy improves with respect to smoothness in the regime $\alpha \leq m(m-1)/2$; however, in the regime $\alpha > m(m-1)/2$, the accuracy becomes a constant with respect to smoothness. The result implies a polynomial of degree $\approx (m-2)(m+1)/2$ is sufficient for accurate recovery of order-m tensors, whereas higher degree brings no further benefits. This full picture of error dependence, to our best knowledge, is new to the literature.

5 Computational limits and polynomial-time algorithms

We should point out that the least-squares estimator $(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}})$ requires exponential-time algorithms, even in the simple matrix case [16]. Specifically, in this section we show the non-existence of *polynomial-time algorithms* with rate (2) under the computational hardness conjecture for hypergraphic planted clique (HPC) [24]. The computational limit implies the necessity of extra assumptions to achieve the optimal rate (2) for any polynomial algorithm.

5.1 Computational limits under HPC detection conjecture

The hypergraphic planted clique detection conjecture plays an important role in constructing the computational limits of our problem. We briefly introduce the HPC hardness conjecture.

Consider an m-uniform hypergraph $G = (V, E)$, where V is a set of vertices, and E is a set of hyperedges. An Erdös-Rényi random hypegraph, denoted by $\mathcal{G}_m(d, 1/2)$, is a random m-uniform hypergraph with d vertices and probability $1/2$ for each of the hyperedge connections. The hypergraphic planted clique (HPC) with clique size $\kappa > 0$, denoted by $\mathcal{G}_m(d, 1/2, \kappa)$, is generated from Erdös-Rényi random hypegraph in the following way. First we generate an Erdös-Rényi random hypegraph from $\mathcal{G}_m(d, 1/2)$. Then we independently pick κ vertices with uniform probability from d vertices. Finally, we obtain the $\mathcal{G}_m(d, 1/2, \kappa)$

by including only the hyperedges whose vertices all belong to the picked τ vertices. The HPC detection refers to the hypothesis testing problem,

$$H_0 : G \sim \mathcal{G}_m(d, 1/2) \quad \text{v.s.} \quad H_1 : G \sim \mathcal{G}_m(d, 1/2, \kappa). \tag{13}$$

The earlier work [24] presents the following hardness conjecture for testing (13).

Conjecture 1 (HPC detection conjecture [24]). *Consider the HPC problem in* (13) *and let* $m \geq 2$ *is be fixed integer. Suppose* $\limsup_{d\to\infty} \log \kappa / \log \sqrt{d} \leq 1 - \tau$, *for any* $\tau > 0$. *Then, for any polynomial-time test sequence* $\{\phi\}_d : G \mapsto \{0, 1\}$, *we have*

$$\liminf_{d\to\infty} \{\mathbb{P}_{H_0}(\phi(G) = 1) + \mathbb{P}_{H_1}(\phi(G) = 0)\} \geq \frac{1}{2},$$

Now we construct the computational lower bound based on Conjecture 1.

Theorem 3 (Computational lower bound). *Assume Conjecture 1 holds. Define the relaxed smooth tensor class* $\mathcal{P}_{\text{rel}} = \{\Theta : \Theta$ *is generated from* (5) *with fixed latent variables* $\{x_i\}_{i=1}^d$ *and* $f \in \mathcal{F}(\alpha, L)\}$. *Consider the Gaussian model* (3) *with* $\alpha > m/2$. *There exists no polynomial algorithm that achieves the statistical optimal convergence Rate*(d)*; i.e.,*

$$\frac{1}{\text{Rate}(d)} \inf_{\hat{\Theta} \text{ Polynomial-time}} \sup_{\Theta \in \mathcal{P}_{\text{rel}}} \text{MSE}(\hat{\Theta}, \Theta) \to \infty, \quad \text{as } d \to \infty. \tag{14}$$

Theorem 3 shows the *impossibility* of polynomial-time estimator to achieve the optimal statistical rate in the general model. The intuition in the proof is to show the best bound for *polynomial-time* tensor estimation as $d^{-m/2}$, in the absence of extra model structures. The condition $\alpha > m/2$ is a technical assumption to facilitate the proof. Theorem 3 is not a weakness of our proposed estimator; rather, (14) reveals the non-avoidable statistical-computational gap as a nature of the smooth tensor estimation problem. The fact has motivated us to introduce extra model structure to fill the gap.

5.2 Borda count algorithm

The earlier section has shown the impossibility of polynomial-time algorithms in the general model. In this section, we restrict ourself to a sub-model with extra monotonicity structures; this structure makes polynomial-time algorithm possible. We introduce a notion of marginal monotonicity for the generative functions.

Definition 2 (Marginal monotonicity)**.** Let $\beta \geq 0$ be a non-negative constant. A function $f\colon [0,1]^m \to \mathbb{R}$ is called β-monotonic, denoted as $f \in \mathcal{M}(\beta)$, if and only if,

$$\left(\frac{i-j}{d}\right)^{1/\beta} \lesssim g(i) - g(j), \quad \text{for all } j < i \in [d], \tag{15}$$

where we define the *score function* $g(i) = d^{-(m-1)} \sum_{(i_2,\ldots,i_m)\in[d]^{m-1}} f\left(\frac{i}{d}, \frac{i_2}{d}, \ldots, \frac{i_m}{d}\right)$ for $i \in [d]$.

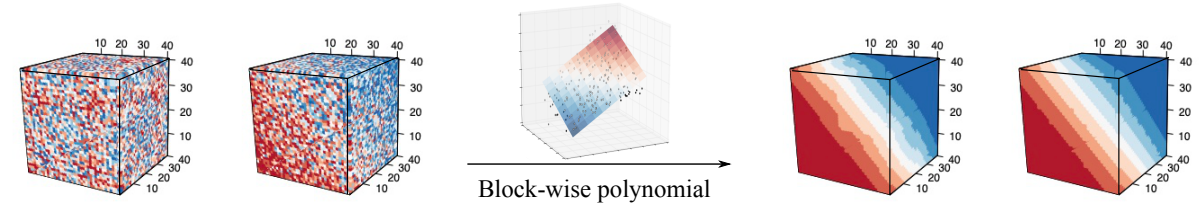
We refer to $\mathcal{F}(\alpha, L) \cap \mathcal{M}(\beta)$ as the monotonic-plus-smooth function class. This class was initially proposed in previous literature of graphons. The work [7] proposes the Lipschitz monotonic function to facilitate the analysis of sorting-merging algorithm for matrix estimation; their setting is a special case of our Definition 2 with $(\alpha, \beta, m) = (1, 1, 2)$. Inspired by earlier work, we consider the similar monotonic-plus-smooth function class $\mathcal{F}(\alpha, L) \cap \mathcal{M}(\beta)$ under general configuration $\{(\alpha, \beta, m) : \alpha > 0,\ 0 < \beta \min(\alpha, 1) \leq 1,\ m \geq 2\}$. Note that the constraint $\beta \min(\alpha, 1) \leq 1$ is due to the natural relationship between joint smoothness and marginal smoothness. A large value of β in (15) implies the steepness of g.

Now we introduce the *Borda count* estimation that consists of two stages. The full procedure is illustrated in Figure 2. Define the empirical score function $\tau\colon [d] \to \mathbb{R}$ as

$$\tau(i) = \frac{1}{d^{m-1}} \sum_{(i_2,\ldots,i_m)\in[d]^m} \mathcal{Y}(i, i_2, \ldots, i_m).$$

1. **Sorting stage:** The sorting stage is to rearrange the observed tensor $\mathcal{Y}$ so that the score function τ of sorted tensor is monotonically increasing. Define a permutation $\hat{\pi}^{\mathrm{BC}}$ such that

$$\tau \circ (\hat{\pi}^{\mathrm{BC}})^{-1}(1) \leq \tau \circ (\hat{\pi}^{\mathrm{BC}})^{-1}(2) \leq \cdots \leq \tau \circ (\hat{\pi}^{\mathrm{BC}})^{-1}(d). \tag{16}$$



(a) Observation (b) Sorted observation (c) Estimated signal (d) True signal

Figure 2: Illustration of Borda count estimation. We first sort tensor entries using the proposed procedure, and then estimate the signal by block-wise polynomial approximation.

Then, we obtain sorted observation $\tilde{\mathcal{Y}} = \mathcal{Y} \circ (\hat{\pi}^{\mathrm{BC}})^{-1}$, illustrated in Figure 2(b).

2. **Block-wise polynomial approximation stage:** Given sorted observation $\tilde{\mathcal{Y}}$, we estimate the signal tensor by block-wise polynomial tensor based on the following optimization,

$$\hat{\Theta}^{\mathrm{BC}} = \underset{\Theta \in \mathscr{B}(k,\ell)}{\arg\min} \|\tilde{\mathcal{Y}} - \Theta\|_F, \tag{17}$$

where $\mathcal{B}(k, \ell)$ denotes the block-k degree-ℓ tensor class in (7). An example of this procedure is shown in Figure 2(c). The estimator $\hat{\Theta}^{\mathrm{BC}}$ depends on two tuning parameters: the number of blocks k and polynomial degree ℓ. The optimal choice of (k^*, ℓ^*) is provided in Theorem 4.

The following theorem ensures the statistical accuracy of the Borda count estimator.

Theorem 4 (Estimation error for Borda count algorithm under marginal monotonicity)**.** *Consider the sub-Gaussian model* (3) *with* $f \in \mathcal{F}(\alpha, L) \cap \mathcal{M}(\beta)$*. Denote a constant* $c(\alpha, \beta, m) := \frac{m(m-1)\beta \min(\alpha,1)}{\max(0,\ 2(m-(m-1)\beta \min(\alpha,1)))}$*. Let* $(\hat{\Theta}^{\mathrm{BC}}, \hat{\pi}^{\mathrm{BC}})$ *be the Borda count estimator with* $\ell^* = \min(\lfloor \alpha \rfloor, \lfloor c(\alpha, \beta, m) \rfloor)$ *and* $k^* \asymp d^{m/(m+2\min(\alpha,\ell^*+1))}$ *in* (17)*. Then, we have*

$$\mathrm{MSE}(\hat{\Theta}^{\mathrm{BC}} \circ \hat{\pi}^{\mathrm{BC}},\ \Theta \circ \pi) \lesssim \begin{cases} d^{-\frac{2m\alpha}{m+2\alpha}} & \text{when } \alpha < c(\alpha, \beta, m), \\ \left(\frac{\log d}{d^{m-1}}\right)^{\beta \min(\alpha,1)} & \text{when } \alpha \geq c(\alpha, \beta, m). \end{cases} \tag{18}$$

The estimation bound (18) comes from the approximation error (Proposition 1), nonparametric error (Theorem 1), and permutation error (Lemma 3).

Remark 4 (Sufficiently smooth tensors)**.** When the generative function is infinitely smooth ($\alpha = \infty$) with Lipschitz monotonic score ($\beta = 1$), our estimation error (18) becomes

$$\mathrm{MSE}(\hat{\Theta}^{\mathrm{BC}} \circ \hat{\pi}^{\mathrm{BC}},\ \Theta \circ \pi) \lesssim d^{-(m-1)} \log d, \tag{19}$$

under the choice of degree and block number

$$\ell^* = (m-2)(m+1)/2 \quad \text{and} \quad k^* \asymp d^{\frac{m}{m+2(\ell^*+1)}}. \tag{20}$$

Now, we compare the rate (19) with the classical low-rank estimation [33, 37, 20]. The low-rank tensor model with a constant rank is known to have MSE rate $\mathcal{O}(d^{-(m-1)})$ [33]. Our infinitely smooth tensor model achieves the same rate up to the negligible log term.

Hyperparameter tuning. Our algorithm has two tuning parameters (k, ℓ). The theoretically optimal choices of (k, ℓ) are given in Theorems 1 and 4. In practice, since model configuration is unknown, we search (k, ℓ) via cross-validation. Based on our theorems, a polynomial of degree $\ell^* = (m-2)(m+1)/2$ is sufficient for accurate recovery of order-m tensors, whereas higher degree brings no further benefit. The practical impacts of hyperparameter tuning are investigated in Section 6.

5.3 Possible relaxation of monotonicity

We emphasize that our monotonicity assumption should be interpreted *up to permutation* in light of (3). The permutation relaxes the extent of stringency in the monotonic assumptions. In the univariate case ($m = 1$), *every smooth function is monotonic-plus-smooth up to permutations.* In particular, $f \in \mathcal{F}(1, L)$ implies the existence of permutation π such that $(f \circ \pi) \in \mathcal{F}(1, L) \cap \mathcal{B}(1)$. The monotonicity comes for free in this case, because the descending sorting, as a permutation, does not deteriorate the 1-d smoothness. For general

order m, the monotonicity imposes moderate constraints on the marginal structure. We provide two examples to illustrate flexibility allowed in our models.

Example 4 (Free monotonicity inherited from smoothness)**.** Consider a quadratic function $f(x, y, z) = (x - 0.5)^2 + yz$. Although the function f is non-monotonic, we can show that f is nearly monotonic-plus-smooth up to permutations; i.e., f can be identified by $\bar{f} \circ \pi$ for some permutation π and $\bar{f} \in \mathcal{F}(\alpha, L) \cap \mathcal{B}(\beta)$ up to a negligible perturbation. See appendix B for the expression of $(\pi, \bar{f})$. Therefore, our Borta count algorithm is applicable to f.

Example 5 (Decomposable monotonicity)**.** Let $R \in \mathbb{N}_+$ be a constant, and $\{g_{r,i}(\cdot) \colon [0,1] \to \mathbb{R}\}$ be a set of 1-d smooth functions for $(r, i) \in [R] \times [m]$. Then, all decomposable smooth functions of the form $f(x_1, \ldots, x_m) = \sum_{r \in [R]} g_{r,1}(x_1) \cdots g_{r,m}(x_m)$ are monotonic-plus-smooth up to permutations. In particular, low-rank tensors with smooth factors are also monotonic-plus-smooth up to permutations.

We show below that the additional monotonic assumptions do not change the minimax rate under Lipschitz (or equivalently, 1-smooth) condition.

Theorem 5 (Statistical minimax lower bound for Lipschitz and monotonic functions)**.** *Consider Lipschitz monotonic functions such that $f \in \mathcal{F}(1, L) \cap \mathcal{M}(1)$. Define the Lipschitz monotonic tensor class $\mathcal{P}_{\text{mon}} = \{\Theta \colon \Theta$ is generated from (4) and $f \in \mathcal{F}(1, L) \cap \mathcal{M}(1)\}$. Then, the estimation problem based on Gaussian model (1) obeys the minimax lower bound*

$$\inf_{(\hat{\Theta}, \hat{\pi})} \sup_{\Theta \in \mathcal{P}_{\text{mon}}, \pi \in \Pi(d,d)} \mathbb{P}\left(\text{MSE}(\hat{\Theta} \circ \hat{\pi},\ \Theta \circ \pi) \gtrsim \text{Rate}(d) \right) \geq 0.8. \tag{21}$$

This result implies that the extra monotonicity assumption renders no changes to the minimax optimal rate of the problem, in the special Lipschitz situation when $\alpha = \beta = 1$. We conjecture that similar results hold true when the nonparametric error dominates the

rate (e.g., $\alpha < c(\alpha, \beta, m)$ in (18)). For general (α, β), the optimality is unknown; we discuss the proof challenges in Appendix E.

Remark 5 (Other monotonicity assumptions)**.** One may also consider other assumptions such as isotonic functions [26]. Define an isotonic function class $\mathcal{M}$

$$\mathcal{M} = \{f \colon [0,1]^m \to \mathbb{R} \;\big|\; f(x_1, \ldots, x_m) \le f(x_1', \ldots, x_m') \text{ when } x_i \le x_i' \text{ for } i \in [m]\}. \quad (22)$$

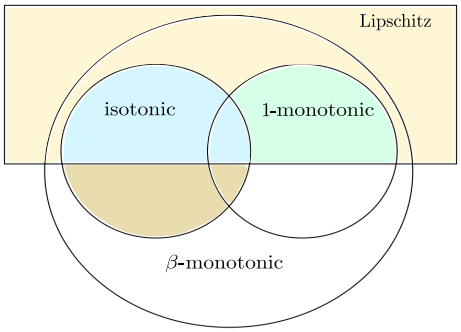
The isotonic functions (22) concerns the *joint* monotonicity, where as our monotonicity (2) concerns the *marginal* monotonicity. The latter is a weaker assumption. The isotonic functions belong to $\mathcal{M}(0)$ based on our Definition 2. Therefore, all our upper bounds apply to isotonicity functions, although such extension is not necessarily sharp. The extension of sharp bounds in Theorems 4-5 to isotonic functions can be found in Appendix C.6.

We summarize the performance of our Borda count algorithm under various model assumptions in Figure 3. The minimax lower bounds are also illustrated for comparison. We find that the Borda count algorithm achieves computational and statistical optimality in the region $\mathcal{F}(1, L) \cap \mathcal{M}$, with $\mathcal{M}$ being either isotonic or 1-monotonic. The optimality may not be attained in the absence of monotonicity, i.e., in the region $\mathcal{F}(\alpha, L)/\mathcal{M}$. Although the statistical-computational gap is non-avoidable in general (Theorem 3), the additional monotonicity assumptions fill in the gap in several cases.

6 Numerical analysis

6.1 Synthetic data

We simulate order-3 d-dimensional tensors based on the permuted smooth tensor model (4). Both symmetric and non-symmetric tensors are investigated. The symmetric tensors are



	Minimax lower bound	Computational upper bound (Borda Count)
	$d^{-\frac{2m}{m+2}} \vee \left(\frac{\log d}{d^{m-1}}\right)$ (Theorem 2)	Unknown
	$d^{-\frac{2m}{m+2}} \vee \left(\frac{\log d}{d^{m-1}}\right)$ (Theorem 5)	$d^{-\frac{2m}{m+2}} \vee \left(\frac{\log d}{d^{m-1}}\right)$ (Theorem 4)
	$d^{-\frac{2m}{m+2}} \vee \left(\frac{\log d}{d^{m-1}}\right)$	$d^{-\frac{2m}{m+2}} \vee \left(\frac{\log d}{d^{m-1}}\right)$
	d^{-1} [31]	$\frac{\log^{5/2} d}{d}$ [31]

Figure 3: Comparison of statistical lower bound and computational upper bound of Borda count algorithm under special case $\alpha = 1$. The right table summarizes the statistical and computational rates corresponding to each colored region in the Venn diagram.

generated based on functions f in Table 2, and the non-symmetric set-up is described in Appendix A.2. The generative functions involve compositions of operations such as

Model ID	$f(x, y, z)$	CP rank	Tucker rank	$\geq (\alpha, \beta)$	Isotonic
1	xyz	1	$(1, 1, 1)$	$(\infty, 1)$	$\surd$
2	$(x + y + z)/3$	3	$(2, 2, 2)$	$(\infty, 1)$	$\surd$
3	$(1 + \exp(-(3x^2 + 3y^2 + 3z^2))^{-1}$	9	$(4, 4, 4)$	$(\infty, 1/2)$	$\surd$
4	$\log(1 + \max(x, y, z))$	≥ 100	$\geq (50, 50, 50)$	$(1, 1)$	$\surd$
5	$\exp\left(-\max(x, y, z) - \sqrt{x} - \sqrt{y} - \sqrt{z}\right)$	≥ 100	$\geq (50, 50, 50)$	$(1/2, 1)$	$\surd$

Table 2: Smooth functions in simulation. We define the numerical CP/Tucker rank as the minimal rank r for which the relative approximation error is below 10^{-4}. The reported rank in the table is estimated from a $100 \times 100 \times 100$ signal tensor generated by (4).

polynomial, logarithm, exponential, square roots, etc. Notice that considered functions cover a reasonable range of model complexities from low rank to high rank. Two types of noise are considered: Gaussian noise and Bernoulli noise. For the Gaussian model, we simulate continuous-valued tensors with i.i.d. noises drawn from $N(0, 0.5^2)$. For the Bernoulli model, we generate binary tensors $\mathcal{Y}$ using the success probability tensor $\Theta \circ \pi$. The permutation π

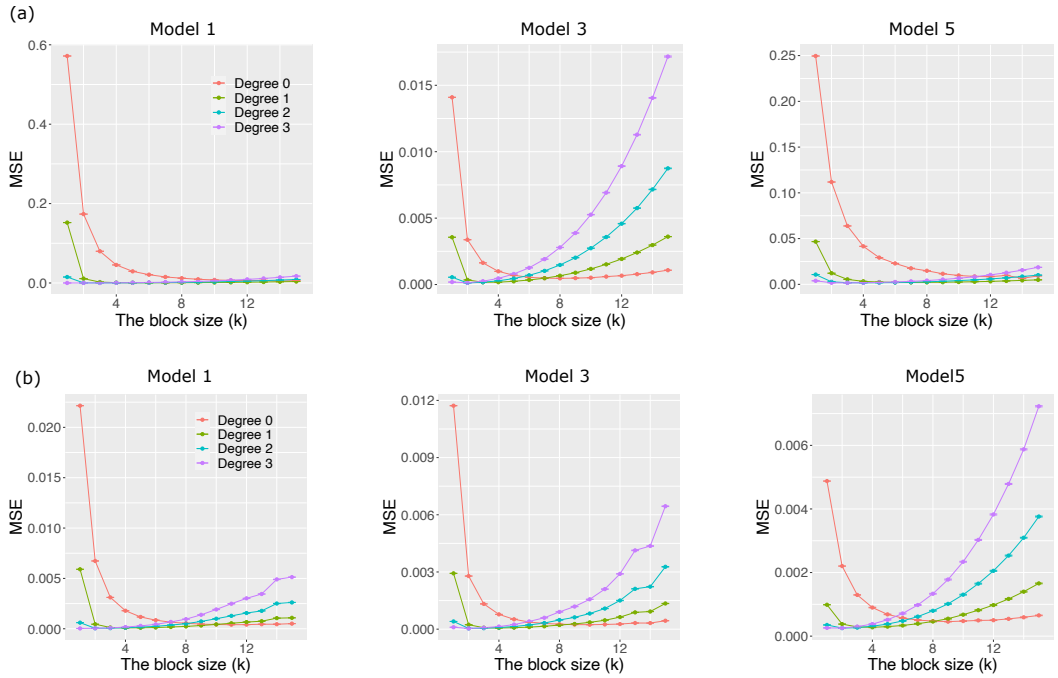


Figure 4: MSE versus the number of blocks based on different polynomial approximations. Columns 1-3 consider the Models 1, 3, and 5 respectively. Panel (a) is for continuous tensors, whereas (b) is for the binary tensors.

is randomly chosen. For space consideration, only results for Models 1, 3, and 5 are presented in the main paper. The rest is presented in Appendix A.1. We first examine impacts of model complexity to estimation accuracy. We then compare Borda count estimation with alternative methods under a range of scenarios. Extensions to non-symmetric tensors and extra simulation results are provided in Appendix A.

Impacts of the number of blocks, tensor dimension, and polynomial degree. The first experiment examines the impact of the block number k and degree of polynomial ℓ for the approximation. We fix the tensor dimension $d = 100$, and vary the number of blocks $k \in \{1, \ldots, 15\}$ and polynomial degree $\ell \in \{0, 1, 2, 3\}$. Figure 4 demonstrates the trade-off in accuracy determined by the number of groups for each polynomial degree. The results confirm our bias-variance analysis in Theorem 1. While a large block number k

provides less biased approximation, this large k renders the signal tensor estimation difficult within each block due to small sample size. In addition, we find that degree-2 polynomial approximation gives the smallest MSE among considered polynomial approximations for models 1-3. By Remark 4, plugging $(\alpha, m) = (\infty, 3)$ in (20) gives the theoretical choice $(k^*, \ell^*) = (d^{7/3}, 2)$. The results are consistent with our simulation.

The second experiment investigates the impact of the tensor dimension d for various polynomial degrees. We vary the tensor dimension $d \in \{10, \ldots, 100\}$ and polynomial degree $\ell \in \{0, 1, 2, 3\}$ in each model configuration. We set optimal number of blocks as the one that gives the best accuracy. Figure S1 compares the estimation errors among different polynomial approximations. The result verifies that the degree-2 polynomial approximation performs the best under the sufficient tensor dimension, which is consistent with our theoretical results. We emphasize that this phenomenon is different from the matrix case where the degree-0 polynomial approximation gives the best results [16, 19].

Comparison with alternative methods. We compare our method (**Borda count**) with several popular alternative methods.

- SVD (**Spectral**) [36] performs universal singular-value thresholding [8] on unfolded tensors.
- Least-squares estimation (**LSE**) [17] uses spectral k-means to approximately solve the optimization problem (8) with constant block approximation ($\ell = 0$).
- Least-squares estimation (**BAL**) [1] uses count-based statistics to approximately solve the optimization problem (8) with constant block approximation ($\ell = 0$). This algorithm is only available for binary observations, so we only use it for comparison under Bernoulli model.

We choose degree-2 polynomial approximation as our theorems suggested, and vary tensor dimension $d \in \{10, \ldots, 100\}$ under each model configuration. Hyperparametes are set to achieve the best performance in the outputs. Possible hyperparameters are the block

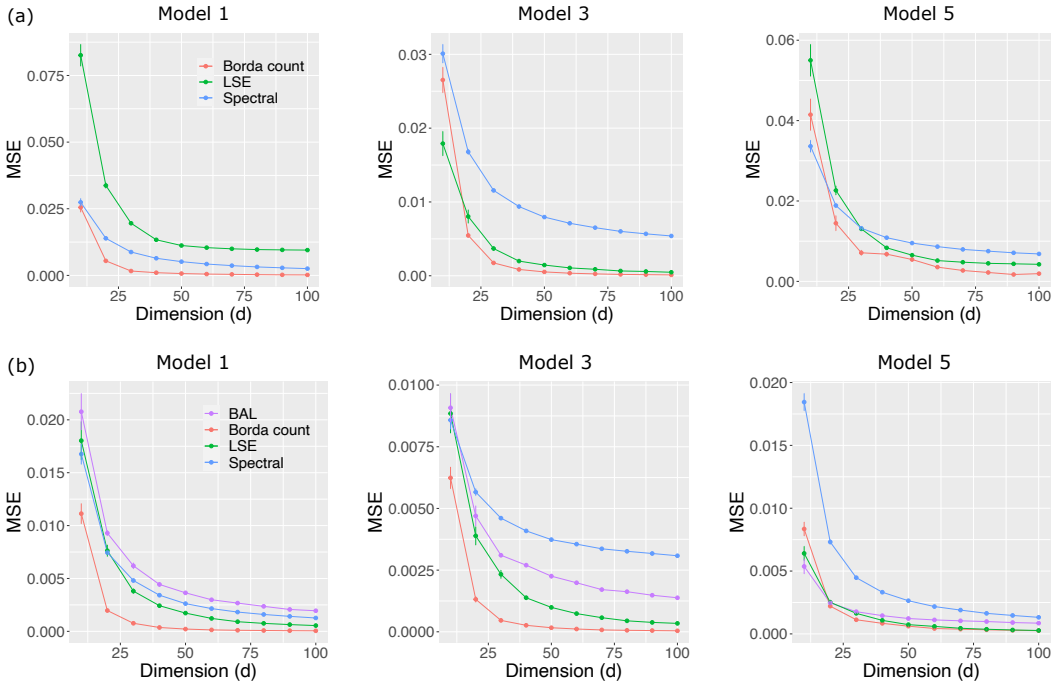


Figure 5: MSE versus the tensor dimension based on different estimation methods. Columns 1-3 consider the Models 1, 3, and 5 in Table 2 respectively. Panel (a) is for continuous tensors, whereas (b) is for the binary tensors.

number for **Borda count**, **LSE** and **BAL**, and the singular-value threshold for **Spectral**.

Figure 5 shows that our algorithm **Borda Count** achieves the best performance in all scenarios as the tensor dimension increases. The poor performance of **Spectral** can be explained by the loss of multilinear structure in the tensor unfolding procedure. The sub-optimality of **LSE** and **BAL** is possibly due to its limits in both statistics and computations. Statistically, constant block approximation results in sub-optimal rates compared to polynomial approximation. Computationally, the least-squares optimization (8) is computationally unstable. Figure 6 displays true signal tensors of three models and corresponding observed tensors of dimension $d = 80$ with Gaussian noise. We use oracle permutation π to obtain the estimated signal tensor from the estimated permuted signal tensor $\hat{\Theta} \circ \hat{\pi}$ for the better visualizations. We see that our **Borda count** achieves the best signal recovery, thereby

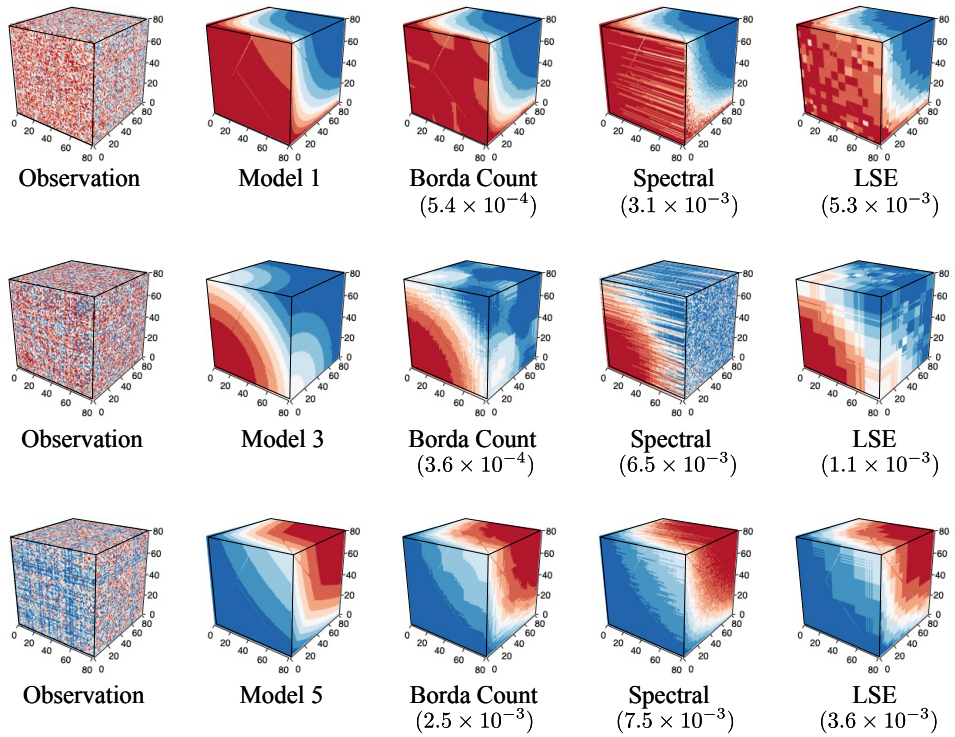


Figure 6: Performance comparison among different methods. The observed data tensors, true signal tensors, and estimated signal tensors are plotted for Models 1, 3 and 5 in Table 2 with fixed dimension $d = 80$. Numbers in parenthesis indicate the mean squared error.

supporting the numerical results in Figure 5.

6.2 Applications to Chicago crime data

The Chicago crime dataset consists of crime counts reported in the city of Chicago, ranging from January 1st, 2001 to December 11th, 2017. The observed tensor is an order-3 tensor with entries representing the log counts of crimes from 24 hours, 77 community areas, and 32 crime types. We apply our Borda count method to Chicago crime dataset. Because the data tensor is non-symmetric, we allow different number of blocks across the three modes. Cross validation result suggests the $(k_1, k_2, k_3) = (6, 4, 10)$, representing the block number for crime hours, community areas, and crime types, respectively.

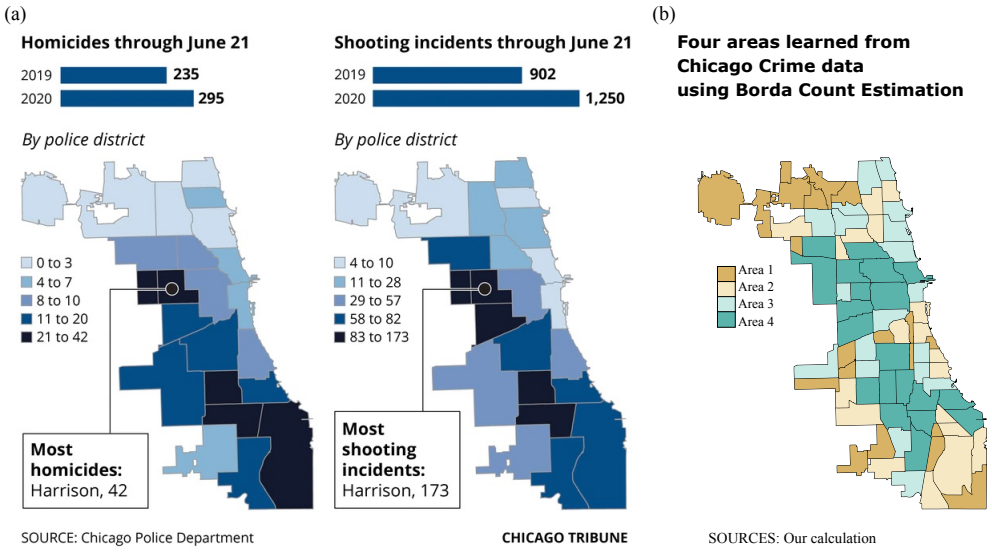


Figure 7: Chicago crime maps. Figure(a) is the benchmark map based on homicides and shooting incidents in community areas in Chicago [18]. Figure(b) shows the four clustered areas learned from 32 crime types using our method.

We first investigate the four community areas obtained from Borda count algorithm. Figure 7(b) shows the four areas overlaid on the Chicago map. Interestingly, we find that the clusters are consistent with actual locations, even though our algorithm did not take any geographic information such as longitude or latitude as inputs. In addition, we compare the cluster patterns with benchmark maps based on homicides and shooting incidents in Chicago shown in Figure 7(a). Our clusters share similar geographical patterns with Figure 7(a). The results demonstrate the power of our approach in detecting patterns from tensor data.

Then, we examine the denoised signal tensor obtained from our method and analyze the trends between crime types and crime hours by the four areas in Figure 7(b). Figure 8 shows the averaged log counts of crimes according to crime types and crime hours by four areas. We find that the major difference among four areas is the crime rates. Area 4 has the highest crime rates, and the crime rates monotonically decrease from Area 4 to Area 1. The variation in crime rates across hour and type, nevertheless, exhibits similarity among

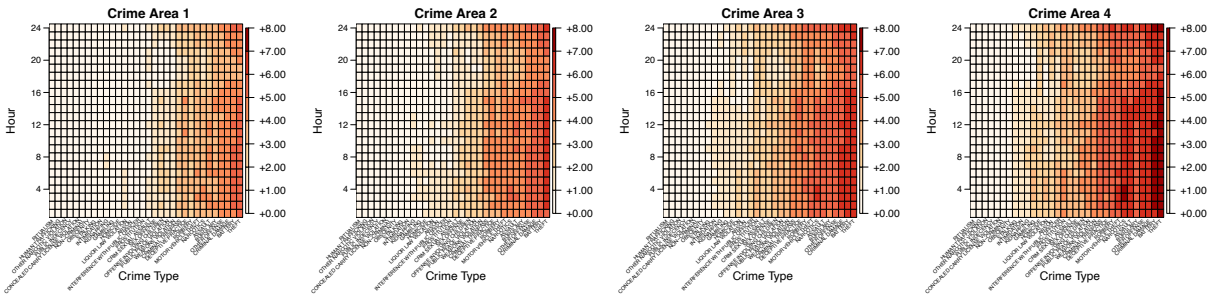


Figure 8: Averaged log counts of crimes according to crime types, hours, and the four areas estimated by our Borda count algorithm. We plot the estimated signal tensor entries averaged within four areas in the heatmap.

the four areas. Figure 8 shows that the number of crimes increases hourly from 8 p.m., peaks at night hours, and then drops to the lowest at 6 p.m. The identified pattens among the four community areas highlight the applicability of our method in real data.

Finally, we compare the prediction performance based on constant block model and our permuted smooth tensor model. Notice that constant block model uses $\ell = 0$ approximation, whereas our permuted smooth tensor model uses $\ell = 2$ approximation. We found that the mean squared prediction errors for our model vs. constant block model are 0.283 (0.006) vs. 0.399 (0.009), respectively. Here, the reported prediction errors are averaged over five runs of cross-validation, with standard errors in parentheses. The block number (k_1, k_2, k_3) with best prediction performance is $(6, 4, 10)$ for our models, and $(7, 11, 10)$ for constant block models. We see that the permuted smooth tensor model substantially outperforms the classical constant block models.

7 Conclusion and Discussions

We have presented a suite of statistical theory, estimation methods, and data applications for permuted smooth tensor models. Two estimation algorithms are proposed with accuracy

guarantees: the (statistically optimal) least-squares estimation and the (computationally tractable) Borda count estimation. In particular, we establish an interesting phase transition phenomenon with respect to the critical smoothness level. We demonstrate that a block-wise polynomial of order $(m-2)(m+1)/2$ is sufficient and necessary for accurate recovery of order-m tensors, in contrast to earlier beliefs on constant block approximation. Experiments demonstrate the effectiveness of both theoretical findings and algorithms.

We discuss several possible extensions from our work. One limitation of our model is that we consider the 1-dimensional latent space embedding. The extension from1-dimensional latent space model to general dimensional latent model is analogous to the extension from the block model to the mixed membership model. Our parallel work [21] considers the general dimensional latent variable model, by assuming a set of s-dimensional vectors $\boldsymbol{a}_{i_k}^{(k)} \in \mathbb{R}^s$ with $s \geq 1$ and a latent function $f\colon [0,1]^s \times \cdots \times [0,1]^s \to \mathbb{R}$ such that

$$\Theta(i_1, \ldots, i_m) = f(\boldsymbol{a}_{i_1}^{(1)}, \ldots, \boldsymbol{a}_{i_m}^{(m)}), \text{ for all } (i_1, \ldots, i_m) \in [d_1] \times \cdots \times [d_m].$$

This generalization extends the latent permutation $\pi \in \Pi(d, d)$ to the set of latent vectors in $[0,1]^s$. However, we find that this extension is not free. We need a stronger analytic function class with ∞-smoothness for the theoretical analysis. Compared to this current paper, the analysis of analytic functions uses different techniques and yields new results of its own. We refer readers to [21] for independent interest.

Another limitation of our algorithm is the requirement of hyperparameter tuning. There is a vast literature on nonparametric estimation that focuses on adaptivity. For example, spatially adaptive methods have been developed in the contexts of wavelets [12], splines [25], and trend filtering [30]; tuning-free algorithms have been proposed for several shape-constrained functions [9, 13, 3]; see [6] for a review. Our work is orthogonal to these advances, and in principle we can combine these tools in our tensor estimation. In this

paper, we choose standard polynomial algorithm because of its simplicity. The parsimony leads to an easier analysis on the critical smoothness level $(m-2)(m+1)/2$. Exploiting various nonparametric techniques for tensor models warrants future research.

References

[1] Balasubramanian, K. (2021). Nonparametric modeling of higher-order interactions via hypergraphons. *Journal of Machine Learning Research 22*, 1–25.

[2] Baltrunas, L., M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, and R. Schwaiger (2011). InCarMusic: Context-aware music recommendations in a car. In *International Conference on Electronic Commerce and Web Technologies*, pp. 89–100. Springer.

[3] Bellec, P. C. (2018). Sharp oracle inequalities for least squares estimators in shape restricted regression. *The Annals of Statistics 46*(2), 745–780.

[4] Bi, X., A. Qu, and X. Shen (2018). Multilayer tensor factorization with applications to recommender systems. *The Annals of Statistics 46*(6B), 3308–3333.

[5] Bickel, P. J. and A. Chen (2009). A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences 106*(50), 21068–21073.

[6] Cai, T. T. (2012). Minimax and adaptive inference in nonparametric function estimation. *Statistical Science 27*(1), 31–50.

[7] Chan, S. and E. Airoldi (2014). A consistent histogram estimator for exchangeable graph models. In *International Conference on Machine Learning*, pp. 208–216.

[8] Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics 43*(1), 177–214.

[9] Chatterjee, S. and J. Lafferty (2019). Adaptive risk bounds in unimodal regression. *Bernoulli 25*(1), 1–25.

[10] Chen, P., C. Gao, and A. Y. Zhang (2022). Optimal full ranking from pairwise comparisons. *The Annals of Statistics 50*(3), 1775–1805.

[11] Ding, J., Z. Ma, Y. Wu, and J. Xu (2021). Efficient random graph matching via degree profiles. *Probability Theory and Related Fields 179*(1), 29–115.

[12] Donoho, D. L. and I. M. Johnstone (1994). Ideal spatial adaptation by wavelet shrinkage. *biometrika 81*(3), 425–455.

[13] Feng, O. Y., Y. Chen, Q. Han, R. J. Carroll, and R. J. Samworth (2022). Nonparametric, tuning-free estimation of s-shaped functions. *Journal of the Royal Statistical Society Series B: Statistical Methodology 84*(4), 1324–1352.

[14] Flammarion, N., C. Mao, and P. Rigollet (2019). Optimal rates of statistical seriation. *Bernoulli 25*(1), 623–653.

[15] Gao, C., Y. Lu, Z. Ma, and H. H. Zhou (2016). Optimal estimation and completion of matrices with biclustering structures. *Journal of Machine Learning Research 17*(1), 5602–5630.

[16] Gao, C., Y. Lu, and H. H. Zhou (2015). Rate-optimal graphon estimation. *The Annals of Statistics 43*(6), 2624–2652.

[17] Han, R., Y. Luo, M. Wang, and A. R. Zhang (2022). Exact clustering in tensor block model: Statistical optimality and computational limit. *Journal of the Royal Statistical Society Series B: Statistical Methodology 84*(5), 1666–1698.

[18] Jeremy, G. A trying first half of 2020 included spike in shootings and homicides in chicago. *Chicago Tribune, 2020-06-26*.

[19] Klopp, O., A. B. Tsybakov, and N. Verzelen (2017). Oracle inequalities for network models and sparse graphon estimation. *The Annals of Statistics 45*(1), 316–354.

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

[21] Lee, C. and M. Wang (2023). Statistical and computational rates in high rank tensor estimation. *arXiv preprint arXiv:2304.04043*.

[22] Li, Y., D. Shah, D. Song, and C. L. Yu (2019). Nearest neighbors for matrix estimation interpreted as blind regression for latent variable model. *IEEE Transactions on Information Theory 66*(3), 1760–1784.

[23] Lovász, L. (2012). *Large networks and graph limits*, Volume 60. American Mathematical Soc.

[24] Luo, Y. and A. R. Zhang (2022). Tensor clustering with planted structures: Statistical optimality and computational limits. *The Annals of Statistics 50*(1), 584–613.

[25] Mammen, E. and S. Van De Geer (1997). Locally adaptive regression splines. *The Annals of Statistics 25*(1), 387–413.

[26] Pananjady, A. and R. J. Samworth (2022). Isotonic regression with unknown permutations: Statistics, computation and adaptation. *The Annals of Statistics 50*(1), 324–350.

[27] Rigollet, P. and J.-C. Hütter (2015). High dimensional statistics. *Lecture notes for course 18S997 813*(814), 46.

[28] Shah, N., S. Balakrishnan, and M. Wainwright (2019). Low permutation-rank matrices: Structural properties and noisy completion. *Journal of Machine Learning Research 20*, 1–43.

[29] Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics 10*(4), 1040–1053.

[30] Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics 42*(1), 285–323.

[31] Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer Science & Business Media.

[32] Wang, M., J. Fischer, and Y. S. Song (2019). Three-way clustering of multi-tissue multi-individual gene expression data using semi-nonnegative tensor decomposition. *The Annals of Applied Statistics 13*(2), 1103–1127.

[33] Wang, M. and L. Li (2020). Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality. *Journal of Machine Learning Research 21*(154), 1–38.

[34] Wang, M. and Y. Zeng (2019). Multiway clustering via tensor block models. In *Advances in Neural Information Processing Systems*, pp. 713–723.

[35] Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.

[36] Xu, J. (2018). Rates of convergence of spectral methods for graphon estimation. In *International Conference on Machine Learning*, pp. 5433–5442.

[37] Zhang, A. and D. Xia (2018). Tensor SVD: Statistical and computational limits. *IEEE Transactions on Information Theory 64*(11), 7311 – 7338.

[38] Zhao, Y. (2015). Hypergraph limits: a regularity approach. *Random Structures & Algorithms 47*(2), 205–226.