Response Letter

We are grateful to the reviewers for their careful reading and thoughtful comments. Please note that the **Response to Major Comments Summarized by Editor** consists of general comments raised by both reviewers. We address these issues upfront **in pages 1-7 of this letter**.

Response to Major Comments Summarized by Editor

1. *The question about whether the monotonicity assumption for the Borda count algorithm can be weaken.*

Response: Inspired by reviewers' comments, we have improved our results on computational limits. We add a new section 5.1 with a new Theorem 3, which roughly says:

> "There exists *no* polynomial-time algorithms that achieve the minimax statistical bound in the general setting, under the popular computational hardness conjecture on hypergraphic planted clique (HPC) detection and some technical conditions."

The new theory has suggested the need of extra assumptions for polynomial-time algorithms. The new section reads:

> "...We should point out that computing the least-squares optimizer $(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}})$ in (8) generally entails exponential complexity, even in the simple matrix case (Gao et al., 2015). 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) (Luo and Zhang, 2022). The computational limit implies the necessity of extra assumptions to achieve the optimal rate (2) via polynomial algorithms.
>
> ### 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,
>
> $$H_0 : G \sim \mathcal{G}_m(d, 1/2) \quad \text{v.s.} \quad H_1 : G \sim \mathcal{G}_m(d, 1/2, \kappa). \tag{1}$$

The earlier work (Luo and Zhang, 2022) presents the following hardness conjecture for testing (1).

Conjecture 1 (HPC detection conjecture (Luo and Zhang, 2022)). *Consider the HPC problem in* (1) *and let* $m \geq 2$ *is be fixed integer. Suppose*

$$\limsup_{d\to\infty} \log \kappa / \log \sqrt{d} \leq 1 - \tau, \quad \text{for any } \tau > 0.$$

Then, for any polynomial-time test sequence $\{\phi\}_d \colon G \mapsto \{0, 1\}$, *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 0.1 (Computational lower bound under sufficient smoothness). *Assume Conjecture 1 holds. Define the relaxed smooth tensor class*

$$\mathcal{P}_{\text{rel}} = \{\Theta : \Theta \text{ is generated from (10) with fixed latent variables } \{x_i\}_{i=1}^d \text{ and } f \in \mathcal{F}(\alpha, L)\}.$$

Consider the Gaussian tensor model (3) *with* $\alpha > m/2$. *Then, there exists no polynomial algorithm that achieves the statistical optimal convergence Rate*(d). *That is,*

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

Theorem 0.1 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 0.1 is not a weakness of our proposed estimator; rather, it 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.....

We have also updated the Table 2 for fair comparison.

	Pananjady and Samworth (2022)	Balasubramanian (2021)	Li et al. (2019)	**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 (e.g., when $(m, \alpha) = (3, 1)$)	d^{-1}	$d^{-\frac{2m}{m+2}}$ $(d^{-6/5})$	$d^{-\lfloor m/3 \rfloor}$ (d^{-1})	$d^{-\frac{2m\alpha}{m+2\alpha}} \vee d^{-(m-1)}$ $(d^{-6/5})$	$d^{-\frac{2m\alpha}{m+2\alpha}} \vee d^{-(m-1)}$ $(d^{-6/5})$
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.

We have also added a new Appendix C.4 for the proof of Theorem 0.1.

C.4 Proof of Theorem 0.1

The proof of Theorem 0.1 leverages results of hypergraphic planted clique and constant higher-order clustering problems. We first briefly explain the constant higher-order clustering problems. We then prove the main result.

C.4.1 Constant higher-order clustering and computational lower bound

Let $\boldsymbol{k} = (k_1, \ldots, k_m)$ and $\boldsymbol{d} = (d_1, \ldots, d_m)$. We introduce the constant high-order clustering (CHC) problem (Luo and Zhang, 2022). Consider a data tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ generated from the signal plus noise model

$$\mathcal{Y} = \Theta + \mathcal{E}, \tag{2}$$

where the entries in $\mathcal{E}$ are i.i.d. drawn from Gaussian distribution, and the signal tensor Θ contains a constant planted structure:

$$\Theta \in \Theta_{\mathrm{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda) := \{\lambda' \mathbb{1}_{I_1} \otimes \cdots \otimes \mathbb{1}_{I_m} \colon |I_i| = k_i, \text{ for all } i \in [m], \ \lambda' \geq \lambda\}.$$

Here $I_i \subset [d_i]$ denotes a subset of indices, $|\cdot|$ denotes the cardinality of the set, $\mathbb{1}_{I_i}$ is the d_i-dimensional indicator vector such that $(\mathbb{1}_{I_i})_j = 1$ if $j \in I_i$ and 0 otherwise. The CHC detection problem is to test the following hypothesis based on the observed tensor $\mathcal{Y}$,

$$H_0 \colon \Theta = 0 \quad \text{v.s.} \quad H_1 \colon \Theta \in \Theta_{\mathrm{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda). \tag{3}$$

The following proposition provides the asymptotic regime for impossible polynomial-time detection of CHC under Conjecture 1. This proposition plays important role to prove our Theorem 0.1.

Proposition 1 (Theorem 15 in (Luo and Zhang, 2022))**.** *Consider CHC detection problem in* (3) *in the Gaussian noise model* (2) *under the asymptotic regime $d \to \infty$ satisfying*

$$d = d_1 = \cdots = d_m, \quad k = k_1 = \cdots = k_m = d^{\delta}, \quad \lambda = d^{-\gamma},$$

with $0 \leq \delta \leq 1$ and $\gamma > (m\delta - m/2) \vee 0$. Then, under Conjecture 1, for any polynomial-time test sequence $\{\phi\}_d \colon \mathcal{Y} \mapsto \{0, 1\}$, we have

$$\liminf_{d \to \infty} \left\{ \mathbb{P}_{\mathcal{H}_0}(\phi(\mathcal{Y}) = 1) + \sup_{\Theta \in \Theta_{\mathrm{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda)} \mathbb{P}_{\Theta}(\phi(\mathcal{Y}) = 0) \right\} \geq \frac{1}{2}.$$

C.4.2 Proof of Theorem 0.1

Proof of Theorem 0.1. Assume that the true signal $\Theta \in \mathbb{R}^{d \times \cdots \times d}$ has constant planted structure with a given $\boldsymbol{k} = (k, \ldots, k)$ such that

$$\Theta \in \Theta_{\mathrm{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda) = \{\lambda \mathbb{1}_I \otimes \cdots \otimes \mathbb{1}_I \colon |I| = k\}.$$

We have $\Theta_{\mathrm{CHC}} \subset \mathcal{P}_{\mathrm{rex}}$, because we can set the infinitely smooth function $f : [0, 1]^m \to [0, 1]$ by

$$f(x_1, \ldots, x_m) = \lambda \prod_{i \in [m]} x_i. \tag{4}$$

Then, there is one-to-one correspondence between tensors in $\Theta_{\text{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda)$ and tensors generated by the above f under the choice $x_i = \mathbb{1}_{i \in I}$ for all $i \in [m]$.

We consider the regime where polynomial-time solvable test is impossible based on Proposition 1. We set $\delta = 1/2$, $k = c_1 d^{\delta}$, and $\lambda = c_2 d^{-\gamma}$ for any fixed $\gamma \in (0, \frac{2\alpha - m}{m})$, so that any polynomial-time test sequence ϕ satisfies

$$\liminf_{d \to \infty} \left\{ \mathbb{P}_{\mathcal{H}_0}(\phi(\mathcal{Y}) = 1) + \sup_{\Theta \in \Theta_{\text{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda)} \mathbb{P}_{\Theta}(\phi(\mathcal{Y}) = 0) \right\} \geq \frac{1}{2}.$$

The choice of λ is possible given that $\alpha > m/2$. Notice that the choice $\lambda \lesssim O(1)$ ensures the function (4) satisfies the definition (1) for all $\alpha > 0$.

We prove by contradiction. Assume that there exists a hypothetical estimator $\hat{\Theta}$ from a polynomial-time algorithm that attains the rate $\text{Rate}(d)$. Specifically, there exists a constant $b > 0$ such that

$$\limsup_{d \to \infty} \frac{1}{\text{Rate}(d)} \sup_{\Theta \in \Theta_{\text{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda)} \frac{1}{d^m} \mathbb{E} \|\hat{\Theta} - \Theta\|_F^2 \leq b. \tag{5}$$

By Markov's inequality, the inequality (5) implies that, when d is sufficiently large, for all $\Theta \in \Theta_{\text{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda)$ and all $u > 0$, we have

$$\|\hat{\Theta} - \Theta\|_F \leq u \sqrt{\text{Rate}(d) d^m}, \tag{6}$$

with probability at least $1 - b/u$. Consider the hypothesis test in (3). We employ the following test

$$\phi(\mathcal{Y}) = \mathbb{1}(\|\hat{\Theta}\|_F \geq u \sqrt{\text{Rate}(d) d^m}).$$

The Type I error of the test ϕ is controlled by

$$\mathbb{P}_0(\|\hat{\Theta}\|_F \geq u \sqrt{\text{Rate}(d) d^m}) = \mathbb{P}_0(\|\hat{\Theta} - \Theta\|_F \geq u \sqrt{\text{Rate}(d) d^m}) \leq b/u.$$

For Type II error, we obtain,

$$\begin{aligned}
\sup_{\Theta \in \Theta_{\text{CHC}}(k, d, \lambda)} \mathbb{P}_{\Theta}(\phi(\mathcal{Y}) = 0) &= \sup_{\Theta \in \Theta_{\text{CHC}}(k, d, \lambda)} \mathbb{P}_{\Theta}(\|\hat{\Theta}\|_F < u \sqrt{\text{Rate}(d) d^m}) \\
&\leq \sup_{\Theta \in \Theta_{\text{CHC}}(k, d, \lambda)} \mathbb{P}_{\Theta}(\|\hat{\Theta} - \Theta\|_F^2 > \|\Theta\|_F^2 - u^2 \text{Rate}(d) d^m) \\
&\overset{(*)}{\leq} \sup_{\Theta \in \Theta_{\text{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda)} \mathbb{P}_{\Theta}(\|\hat{\Theta} - \Theta\|_F^2 > u^2 \text{Rate}(d) d^m) \\
&\overset{(**)}{\leq} b/u.
\end{aligned}$$

The inequality $(*)$ holds because

$$\|\Theta\|_F^2 \geq \lambda^2 k^m = c_1^m c_2^2 d^{\frac{m}{2} - \gamma} \geq 2u^2 \text{Rate}(d) d^m \asymp d^{\frac{m}{2} - \frac{2\alpha - m}{m}}$$

where the last inequality is true under the regime $c_1^m c_2 > 2u^2$. We can always choose constants c_1 and c_2 given the value u. The inequality $(**)$ holds because of the statement (6). Putting Type I and II errors together, we obtain

$$\mathbb{P}_{\mathcal{H}_0}(\phi(\mathcal{Y}) = 1) + \sup_{\Theta \in \Theta_{\text{CHC}}(\boldsymbol{k}, \boldsymbol{d}, \lambda)} \mathbb{P}_{\Theta}(\phi(\mathcal{Y}) = 0) \leq 2b/u < 1/2,$$

for $u > 4b$. This fact contradicts the Proposition 1. Therefore, there is no polynomial-time $\hat{\Theta}$ satisfying (5). □

2. *The question about whether this assumption should lead to higher convergence rates.*

Response: Inspired by reviewer's comments, we have added a new Theorem 0.2 in the main paper. Our new theorem shows that the monotonicity does not lead to a higher convergence rate in many cases.

We quote the new section here:

> "...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.... "
>
> **5.2 Borda count algorithm**
>
> "...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 Chan and Airoldi (2014) 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. ..."
>
> **5.3. Possible relaxation of monotonicity**
>
> We have shown the success of our Borda count algorithm under extra marginal monotonicity. We refer to $\mathcal{F}(\alpha, L) \cap \mathcal{B}(\beta)$ as the monotonic-plus-smooth function class. This class was initially proposed in previous literature of graphons....
>
> 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 1** (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 2** (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 0.2 (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*

$$\mathcal{P}_{\text{mon}} = \{\Theta \colon \Theta \text{ is generated from (4) and } f \in \mathcal{F}(1, L) \cap \mathcal{M}(1)\}.$$

Then, the estimation problem based on Gaussian model (1) obeys the minimax lower bound

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

This result implies that the extra monotonicity assumption renders no changes to the minimax optimal rate of the problem, in the common 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 1 (Other monotonicity assumptions)**.** One may also consider other assumptions such as isotonic functions (Pananjady and Samworth, 2022). Define an isotonic function class $\mathcal{M}$

$$\mathcal{M} = \{f \colon [0,1]^m \to \mathbb{R} \mid f(x_1, \ldots, x_m) \leq f(x_1', \ldots, x_m') \text{ when } x_i \leq x_i' \text{ for } i \in [m]\}. \quad (7)$$

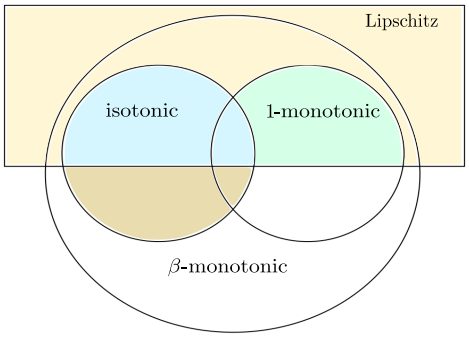
The isotonic functions (9) concerns the *joint* monotonicity, where as our monotonicity (2) concerns the *marginal* monotonicity. The later is a weaker assumption. The isotonic functions belong to $\mathcal{M}(0)$ based on our Definition 2. Therefore, all our upper bounds apply to isotonicity functions, 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 2. 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 0.1), the additional monotonicity assumptions fill in the gap in several cases.

3. *The question about the adaptivity of the algorithm.*

Response: This is an important yet challenging question. We interpret the question as model selection; we address the issue from two perspectives.

– **In practice**, we recommend to search (k, ℓ) via cross-validation in the algorithm. We have added the following subsection in Section 4:



	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 1: Comparison of statistical lower bound and computational upper bound of Borda count algorithm. The right table summarizes the statistical and computational rates corresponding to each colored region in the Venn diagram.

> "... **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.

In Section 6:

> "...The first experiment examines the impact of the block number k and degree of polynomial ℓ for the approximation....Figure 4 demonstrates the trade-off in accuracy determined by the number of groups for each polynomial degree...We find that degree-2 polynomial approximation gives the smallest MSE among considered polynomial approximations for models 1-7. 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."

– **In theory**, the development of tuning-free, adaptive algorithms is important. This general topic applies to all nonparametric problems, not only to our problem per se. We have added the following discussion in Section 7:

> "...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 (Donoho and Johnstone, 1994), splines (Mammen and Van De Geer, 1997), and trend filtering (Tibshirani, 2014); tuning-free algorithms have been proposed for several shape-constrained functions (Chatterjee and Lafferty, 2019; Feng et al., 2022; Bellec, 2018); see (Cai, 2012) 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."

Point-by-point Response to Reviewer 1

This manuscript deals with the interesting problem of estimating an order-m d-dimensional tensor Θ which is α-Holder smooth up to an unknown permutation π–for simplicity it is assumed that the permutation is the same along all modes. The main contribution is two-fold.

First, the author introduce an (exp-time) estimator by minimizing the least-square criterion over all possible partitions of the d points in k block and then estimating the tensor Θ as a piece-wise polynomial degree-ℓ on these blocks. If one suitably tunes k and ℓ, the resulting estimator is shown to achieve the minimax estimation rate.

Second, the authors introduce a simple efficient estimator under the additional β-monotonicity assumption. This assumption allows to almost exactly recover the permutation of the points by simply considering the Borda count of each point. Then, combining this estimated permutation with a piece-wise polynomial estimator allows the authors to recover the desired convergence rate.

The paper is mostly well written. Numerical comparisons illustrate the results. Up to my knowledge, the minimax rate for smooth tensors (up to a permutation) is new. In comparison to the abundant literature for matrix or tensor estimations, the main novel idea is that of using a piece-wise polynomial estimator instead of a piecewise constant vector, which allows the authors to considering α-smooth tensors with $\alpha > 1$.

We greatly appreciate your valuable comments and suggestions. We have carefully addressed all the questions. Note that general comments raised by both reviewers were addressed upfront in **pages 1-7 of this response letter, Response to Major Comments Summarized by Editor**. We will make explicit reference to earlier section when needed.

1. *I find the comparison of the results in Table 1 somewhat misleading. For instance, the row "Polynomial algorithm" may suggest that there exists a Polynomial algorithm for general α-smooth tensors, whereas a polynomial-time estimator is only exhibited under the very restrictive β-monotonicity assumption. In the row "error rate" and columns "α-smoothness" the authors only state the rate for $\alpha = \infty$, whereas the rates for previous works are provided under much weaker regularity assumptions.*

Response: We have updated the table for fair comparison.

	Pananjady and Samworth (2022)	Balasubramanian (2021)	Li et al. (2019)	**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 (e.g., when $(m,\alpha)=(3,1)$)	d^{-1}	$d^{-\frac{2m}{m+2}}$ $(d^{-6/5})$	$d^{-\lfloor m/3 \rfloor}$ (d^{-1})	$d^{-\frac{2m\alpha}{m+2\alpha}} \vee d^{-(m-1)}$ $(d^{-6/5})$	$d^{-\frac{2m\alpha}{m+2\alpha}} \vee d^{-(m-1)}$ $(d^{-6/5})$
Minimax optimality	$\surd$	$\times$	$\times$	$\surd$	$-*$
Polynomial algorithm	$\surd$	$\times$	$\surd$	$\times$	$\surd$

Table 2: Comparison of our results with previous work. For simplicity, we omit the log term in the rate. *Optimality is achieved under extra Lipchitz monotonicity conditions.

We also add new theorem to show the *non-existence* of polynomial-time algorithm achieving the minimax rate without further assumption (see **Point 1 in Response to Major comments Summarized by Editor** for details). We present two estimation algorithms in the revised table: the maximum-likelihood estimation (MLE) and the Borda count estimation. The MLE uses no extra

assumption and improves the previously conjectured optimal rate in the earlier work (Balasubramanian, 2021). The Borda count algorithm achieves efficiency under an extra Lipschitz condition (see **Points 1-2 in Response to Major comments Summarized by Editor** for details).

2. *I have concerns with the proof of the minimax lower bound (Theorem 2). More precisely, I do not understand how one deduces (31) from (29), (30), and Lemma 4 in Page 44. Informally, in (29), the authors reduce their problem to that of estimating a block-constant tensors (up to a permutation) of $(d/2)$ points, while knowing exactly the position of the $d/2$ remaining points. As as consequence, this does not boil down to the setting of Lemma 4, as the statistician also observes all the interactions between the $d/2$ points with known position and the $d/2$ points with unknown position. This additional information could make the problem significantly easier than what is suggested by Lemma 4. Could the authors elaborate on that?*

Response: Thank you for pointing out the gap omitted in the proof. We have added the following lines in the proof to address your concern.

> "The proof of Lemma 2 is constructive and deferred to Section G. The core tensor $\mathcal{S}$ in Lemma 2 has a special pattern of zero's that will be used in the proof of Theorem 2; see the Section G for details....
>
> "...From this observation, we define a sub-domain $I \subset [d]$ such that
>
> $$I = \left(\bigcup_{a=1}^{k} \left[\frac{d(a-3/4)}{k}, \frac{d(a-1/4)}{k} \right] \right) \bigcap [d]. \tag{8}$$
>
> We have that $|I| = d/2$ by definition. Let $\Theta(\mathcal{S}) \in \mathbb{R}^{d \times \cdots \times d}$ denote the tensor induced by f in (34). We use subscript I to denote objects when restricted in the indices set I. For example, $\Theta_I(\mathcal{S}) \in \mathbb{R}^{d/2 \times \cdots \times d/2}$ denotes the sub-tensor with indices in I, and $\|\cdot\|_{F,I}$ denotes the sum of squares over indices in I. Based on (35) and (8), $\Theta_I(\mathcal{S})$ has block structure with the core tensor $\mathcal{S}$.
>
> We use $\Pi(d/2, d/2)$ to denote the set of all permutations on I while fixing indices on $[d] \setminus I$; that is, $\Pi(d/2, d/2) = \{\pi \colon I \to I\} \cong \{\pi \in \Pi(d,d) \colon \pi(i) = i \text{ for } i \in [d] \setminus I\})$. Then, we have
>
> $$\begin{aligned}
> &\inf_{(\hat{\Theta}, \hat{\pi})} \sup_{\pi \in \Pi(d,d)} \mathbb{P}\left(\frac{1}{d^m} \|\hat{\Theta} \circ \hat{\pi} - \Theta(\mathcal{S}) \circ \pi\|_F^2 \geq \varepsilon^2 \right) \\
> \overset{(*)}{=} &\inf_{\hat{\Theta}} \sup_{\pi \in \Pi(d,d)} \mathbb{P}\left(\frac{1}{d^m} \|\hat{\Theta} - \Theta(\mathcal{S}) \circ \pi\|_F^2 \geq \varepsilon^2 \right) \\
> \geq &\inf_{\hat{\Theta}} \sup_{\pi \in \Pi(d,d)} \mathbb{P}\left(\frac{1}{d^m} \|\hat{\Theta} - \Theta(\mathcal{S}) \circ \pi\|_{F,I}^2 \geq \varepsilon^2 \right) \\
> \overset{(**)}{\geq} &\inf_{\hat{\Theta}} \sup_{\pi \in \Pi(d/2,d/2)} \mathbb{P}\left(\frac{1}{(d/2)^m} \|\hat{\Theta}_I - \Theta_I(\mathcal{S}) \circ \pi\|_F^2 \geq 2^m \varepsilon^2 \right) \\
> = &\inf_{\hat{\Theta}} \sup_{z \in \Pi(d/2,k)} \mathbb{P}\left(\frac{1}{(d/2)^m} \|\hat{\Theta}_I - \mathcal{S} \circ z\|_F^2 \geq 2^m \varepsilon^2 \right),
> \end{aligned}$$
>
> where $(*)$ absorbs the estimate $\hat{\pi}$ into the estimate $\hat{\Theta}$, and $(**)$ uses the constructed function (34) and the permutation collections $\Pi(d/2, d/2)$. Based on the construction of $\mathcal{S}$ in Lemma 2, the cross terms in $(\mathcal{S} \circ z)(I, I^c, \ldots)$ are zero. Therefore, we reduce the

problem of estimating $\pi\colon [d] \to [d]$ to estimating $z : I \to [k]$ in the sub-tensor. Applying Lemma 2 to(37) by using $d/2$ in the place of d and $k = d^\delta$ for a constant $\delta > 0$ yields the desired conclusion.

Note that we observe all interactions between the $d/2$ points with known position and $d/2$ points with unknown positions. This addition information occurs only a constant 2^m factor in the final bound, due to zero patterns in the core tensor $\mathcal{S}$ in Lemma 2. Since order of tensor m is fixed, this improvement does not affect the final rate.

3. *First, I am wondering whether the minimax rate should not be faster under this assumption as one could perhaps leverage on this assumption to improve the non-parametric rate.*

Response: Inspired by your comments, we have added a new Theorem 0.2 to provide the minimax rate under monotonicity. It turns out the monotonicity does not improve the rate under Lipschitz condition (equivalently, $\alpha = \beta = 1$ in our setting). The high-level idea is based on the Low and Kang (2002) and the referred proof in Kiefer (1982):

> "(page 364 of Low and Kang (2002))...this extra order constraint (monotone) does not improve optimal rates of convergence..., and thus $d^{-2\alpha/(2\alpha+1)}$ is still the optimal rate of convergence for the minimax risk (for 1-d function estimation)..."

Please see detailed response in **Point 2, Response to Major Comments summarized by Editor (pages 1-7 of this letter)**.

4. *Second, contrary to what is stated in the manuscript, this assumption is not weaker than an isotonic condition. Indeed, isotonic condition prescribe that the difference $g(i)-g(j)$ is non-negative, but not that this different is at least of the order $(i-j)^\beta$.*

Response: The isotonicity belongs to our monotonic model by setting $\beta = \infty$ in $|i-j|^\beta$ based on Definition 2 in the paper[1]. To see this, set $|i-j| = 1$ and $\beta = \infty$. Then $g(i)-g(j) \geq (|i-j|/d)^\beta \to 0$ as $d \to \infty$, reducing to the free-of-order constraint. This is what we meant by the broadness of our framework.

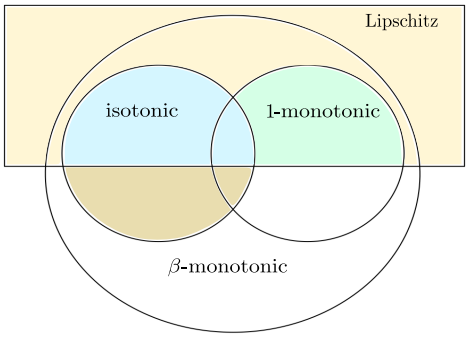
Furthermore, our β-monotonicity model considers the *marginal* monotonicity, while the classical isotonicity model concerns the *joint* monotonicity. The former is a weaker assumption.

We have added the following comment in Section 5.3

> "One may also consider other assumptions such as isotonic functions (Pananjady and Samworth, 2022). Define an isotonic function class $\mathcal{M}$
>
> $$\mathcal{M} = \{f\colon [0,1]^m \to \mathbb{R} \mid f(x_1,\ldots,x_m) \leq f(x_1',\ldots,x_m') \text{ when } x_i \leq x_i' \text{ for } i \in [m]\}. \quad (9)$$
>
> The isotonic functions (9) concerns the *joint* monotonicity, where as our monotonicity (2) concerns the *marginal* monotonicity. The later is a weaker assumption. 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 also added Figure 2 in the main paper to illustrate the relationship between isotonicity and monotonicity. The ∞-monotonicity includes isotonicity, while the 1-monotonicity overlaps with isotonicity. Please see our response to **Point 2, Response to Major Comments summarized by Editor (pages 1-7 of this letter)**.

[1]Our paper uses the notion of $|i-j|^{1/\beta}$, so $\beta = 0$ in our notation.



	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 2: Comparison of statistical lower bound and computational upper bound of Borda count algorithm. The right table summarizes the statistical and computational rates corresponding to each colored region in the Venn diagram.

5. *Are the authors able to address the isotonic case with their procedure?*

Response: Based on our response to #4, the isotonicity is a special case of our monotonicity. Therefore, all our upper bound results equally apply to isotonic functions. The lower bound results, however, need extra work. We summarize our new results in our response to **Point 2, Response to Major Comments summarized by Editor (pages 1-7 of this letter)**.

6. *Minor comments:*

(a) *Page 9, Line 46 "Frobunis"*

(b) *There is a typo in the subscript of (28)*

(c) *Lemma 5 (prosy)*

Response: Thank you for the comments. We have corrected the typos.

Point-by-point Response to Reviewer 2

The paper considers a permuted smooth tensor model (which is essentially an m-dimensional regression problem with unknown correspondences) in the fixed design setup and establishes minimax lower bounds and an estimator that achieves the lower bounds when the true regression function is α-Holder. The main observation is that while prior works considered estimators which were based on piecewise constant approximations, the paper uses piecewise polynomial approximations (with the order of the polynomial picked accordingly). This enables the paper to obtain rates that are minimax optimal. A main issue (as with all prior works) is that the algorithm involves combinatorial search and hence is computationally hard.

The paper then propose an algorithm, based on the so-called Borda counts that provides a linear time algorithm (i.e., linear in dm, the number of entries). However, this requires the more stringent assumption that the true regression function is monotonic. Given this assumption, indeed the algorithm seems a natural algorithm to consider.

I have went over the proofs and they seem to be ok to my knowledge. The paper indeed makes a contribution to this growing literature. Hence, I recommend a major revision that would address the following concerns.

We greatly appreciate your valuable comments and suggestions. We have carefully addressed all the questions. Note that general comments raised by both reviewers were addressed upfront in **pages 1-7 of this response letter, Response to Major Comments Summarized by Editor**. We will make explicit reference to earlier section when needed.

1. *The paper claims the proposed Borda count algorithm is both statistically and computationally efficient. However, the algorithm requires the more stringent assumption of monotonicity. Hence, I personally think this claim is a bit misleading. In fact, it is not clear what is the minimax lower bound for when the truth lies in $\mathcal{F}(\alpha, L) \cap \mathcal{M}(\beta)$.*

Response: Inspired by your comments, we have added a new theorem 5 to show the minimax lower bound for $\mathcal{F}(1, L) \cap \mathcal{M}(1)$. The rate turns out to agree with Borda count algorithm. 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.

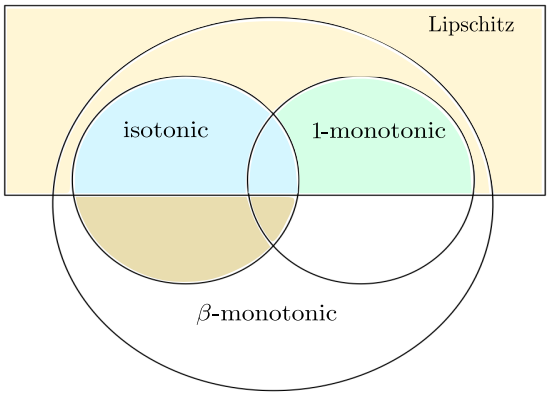
Please see our detailed revision in **Point 2, Response to Major Comments summarized by Editor (pages 1-7 of this letter)**. Throughout the paper, the new conclusion now reads "...Borda count algorithm achieves optimal rate under an extra Lipchitz monotonic condition".

2. *Furthermore, can the authors provide some intuition about this class $\mathcal{F}(\alpha, L) \cap \mathcal{M}(\beta)$?*

Response: We have added the following descriptions in Section 5:

> "...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 (Chan and Airoldi, 2014) 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 $\{(\alpha, \beta, m) : \alpha > 0,\ 0 < \beta \min(\alpha, 1) \leq 1,\ m \geq 2\}$. ...

We added Figure 2 in the main paper for the intuition about function classes. Figure 2 summarizes the relationship among various function classes and corresponding statistical and computational limits. The function class of $\mathcal{F}(\alpha, L)$ concerns the smoothness, while β-monotonicity concerns marginal monotonicity. Please see our response to **Point 2, Response to Major Comments summarized by Editor (pages 1-7 of this letter)**.



	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. The right table summarizes the statistical and computational rates corresponding to each colored region in the Venn diagram.

3. *Do the examples in Model 1 to 5 belong to this class ?*

Response: Yes, all examples in Models 1 to 5 belong to this class. We added the last two columns in Table 2 to present the corresponding model configuration (α, β) for models 1-5.

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 3: Smooth functions in simulation. ...

4. *Adaptivity: Both in Theorems 1 and 3, the result requires knowledge of α. In theorem 3, in addition knowledge about β is required. Could the authors extend their results so that they are adaptive to the choices of α and β?*

Response: This is a great question. We interpret the adaptivity as a model selection problem. Please see **Point 3, Response to Major Comments summarized by Editor (pages 1-7 of this letter)**.

5. *Furthermore, from the experiments, it is not clear how the observed empirical choices of (ℓ^*, k^*) are related to the theoretical choices. The authors mention that these observations are consistent with our theoretical results that the optimal number of blocks and polynomial degree are $(k^*, \ell^*) = (\mathcal{O}(d^{3/7}), 2)$. However this is not clear at all. Can the authors calculate the theoretical choices for models 1 to 5 and show it agrees with the simulation ?*

Response: Thank you for the comments. In the Remark 4 of the main paper, we show that,

...when the generative function is infinitely smooth ($\alpha = \infty$) with Lipschitz monotonic score ($\beta = 1$), ... (the optimal rate is achieved) under the choice of degree and block number

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

In the simulation models, we plug in $(\alpha, m) = (\infty, 3)$ to obtain $(k^*, \ell^*) = (d^{7/3}, 2)$. Therefore, the theoretical choice agrees with the simulation.

We added the following paragraph in the main paper for a clear explanation:

In addition, we find that degree-2 polynomial approximation gives the smallest MSE among considered polynomial approximations for models 1-7. 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.

6. *Also, to my knowledge, the papers [1] and [18] seem to consider random design setup. It is not clear if Table 1 is a fair comparison in this sense. This distinction should perhaps be distinguished right in the introduction rather than in the conclusion section at the end. Furthermore, how is the fixed-design assumption verified for the Chicago crime dataset experiment?*

Response: We revised the Table 1 for fair comparison. The newly added rows are highlighted in blue.

	Pananjady and Samworth (2022)	Balasubramanian (2021)	Li et al. (2019)	**Ours** (MLE)	**Ours** (Borda count)
Model structure	monotonic	Lipschitz	Lipschitz	α-smooth	α-smooth & monotonic
Fixed grid design	$\surd$	$\times$	$\times$	$\times$	$\surd$
⋮	⋮	⋮	⋮	⋮	⋮
Polynomial algorithm	$\surd$	$\times$	$\surd$	$\times$	$\surd$

Table 4: Comparison of our results with previous work...

We have also added the following clarification in Section 1.2:

"...Unlike typical simple hypergraphons where the design points are random, our generative model uses deterministic design points. These two choices lead to different analysis similar as random- vs. fixed-designs in nonparametric regression (Wasserman, 2006; Tsybakov, 2009). The comparison of two approaches will be discussed in Sections 2 and 4."

In Section 2, we have added:

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

$$\Theta(i_1, \ldots, i_m) = f(x_{i_1}, \ldots, x_{i_m}), \quad \text{for all} \quad (i_1, \ldots, i_m) \in [d]^m. \tag{10}$$

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 (10) as the relaxed smooth model; similar model has been developed in the literature of graphons and hypergraphons (Chan and Airoldi, 2014; Gao et al., 2015; Klopp et al., 2017; Balasubramanian, 2021). Our paper will focus on the grid design (4), but we will also discuss the extension to (10) whenever possible as remarks of theorems.

In Section 4, we added:

> **Remark 3** (Phase transition). Our Theorem 1 can be extended to relaxed smooth model (10). In the Appendix F, we show that the relaxed smooth model has the same convergence upper bound Rate(d), with little modification of proofs.

In the conclusion Section, we added:

> We find that our theorems in Section 4 can be extended to random designs; the analysis is provided Appendix F. The extension of theorems in Section 5, however, remain challenging for random designs. A full comparison between the two designs is an interesting question for future research.

We also add a new Appendix F to prove the extension of Theorem 1 and Proposition 1 to random design models.

Similar to other papers (Balasubramanian, 2021; Li et al., 2019), we do not present a statistical test for checking whether the fixed or random design assumption is true in real data application. In our Chicago crime data analysis, we use domain knowledge to validate output results. We believe that verifying nonparametric model assumptions (not restricted to our framework) in real data is generally challenging, and we leave it for future work.

7. *Minor comments: Reference [23] and its citation in Lemma 6 needs to be fixed.*

Response: Thank you for the comments. We have fixed the reference [23] and its citation.

Point-by-point Response to Reviewer 3

This paper proposes a permuted smooth tensor model, where the mean of the observation at $(i_1, i_2, \ldots, i_m)$ *is equal to* $f(i_1/d, i_2/d, \ldots, i_m/d)$*, for a smooth* m*-variate function* f*, subject to an unknown permutation of indices. The authors studied the minimax estimation error when* f *belongs to a Holder class and achieved this rate by a (computationally expensive) least-squares algorithm. The authors also proposed a Borda count algorithm, for the special case where* f *is both smooth and monotone, and showed that it achieves the minimax rate in this case. The algorithm is applied to the Chicago crime count data and produces some interesting results, such as clustering of neighborhoods.*

The paper is very well-written. The methods and main contributions are stated clearly. There are also many interpretations of theory (e.g., why the order should have such a form, how it compared with the error orders in related problems). These interpretations help the readers understand the insights behind the theory, which I really like.

We greatly appreciate your valuable comments and suggestions. We have carefully addressed all the questions. Note that general comments raised by both reviewers were addressed upfront in **pages 1-7 of this response letter, Response to Major Comments Summarized by Editor**. We will make explicit reference to earlier section when needed.

1. *My main concern is that the problem has not yet been fully resolved. I was hoping to see a more complete answer. Without monotonicity, how to develop a polynomial-time algorithm?*

Response: We added new results from two perspectives:

– **In theory,** we have improved our results inspired by your comments. We add a new section 5.1 with a new Theorem 3, which roughly says:

> "There exists *no* polynomial-time algorithms that achieve the minimax statistical bound in the general setting, under the popular computational hardness conjecture on hypergraphic planted clique (HPC) detection and some technical conditions."

Please see **Point 1, Response to Major Comments summarized by Editor (pages 1-7 in this letter)** for more details. Our new theory has suggested the need of extra assumptions for efficient polynomial-time algorithms.

– **In algorithm,** we added several new polynomial-time algorithms in the paper. In particular, the **LSE** in simulation is developed by us, and we treat it as a polynomial-time surrogate to MLE. This algorithm uses spectral method for initialization and constant blocks for approximation (Han et al., 2022). The comparison is provided in Section 6 and the new Figure R2 in this letter. Please see our responses to # 5 and #7 for details.

2. *Under this assumption, the unknown permutation is found by ordering the degrees of nodes. This prevents some common cases such as* f *being a quadratic function, e.g.,*

$$f(x, y, z) = (x - 0.5)^2 + yz. \tag{11}$$

In this case, if we use Borda count to permute and group nodes in mode 1, the algorithm will mistakenly group together those top-ranked and bottom-ranked (in true ordering) nodes. I am wondering whether there exists an algorithm to deal with the general case without monotonicity. It is

okay even if this algorithm doe not have the minimax rate. The current paper does not give any computable algorithm in the general case. This is the main weak-ness of the paper.

Response: In fact, your quadratic function (11) falls precisely in our model class. The function is non-monotonic per se, but it is monotonic *up to permutations.* Our Borda count algorithm works perfectly in your example (11).

We elaborate our response from three perspectives:
– **Your specific example (theory).** We demonstrate the success of our Borda count algorithm for smooth-plus-monotonic functions *up to permutations.* This permutation needs not to be the same as the oracle permutation in the generative model.

We show that Borda count algorithm works perfectly on your example (11). For simplicity, assume that d is an even number. The signal tensor is represented by

$$\Theta(i,j,k) = \left(\frac{i}{d} - 0.5\right)^2 + \left(\frac{j}{d}\right)\left(\frac{k}{d}\right), \quad \text{for all } (i,j,k) \in [d]^3. \tag{12}$$

We construct permutations π_1, π_2, π_3 and a multivariate function $\bar{f}\colon [0,1]^3 \to \mathbb{R}$ such that

$$\Theta(i,j,k) = \bar{f}\left(\frac{\pi_1(i)}{d}, \frac{\pi_2(j)}{d}, \frac{\pi_3(k)}{d}\right) \pm \frac{1}{d^2}, \quad \text{for all } (i,j,k) \in [d]^3. \tag{13}$$

Define the permutations by

$$\pi_1(i) = \begin{cases} 2i - d, & \text{if } i > \frac{d}{2}, \\ d + 1 - 2i, & \text{if } i \le \frac{d}{2}, \end{cases} \quad \text{and} \quad \pi_2(i) = \pi_3(i) = i. \tag{14}$$

Define the function $\bar{f}\colon [0,1]^3 \to \mathbb{R}$ by

$$\begin{aligned} \bar{f}\colon [0,1]^3 &\to [0,1] \\ (x,y,z) &\mapsto \frac{1}{4}x^2 + yz. \end{aligned} \tag{15}$$

One can verify that $\bar{f}$ is monotonic-and-smooth such that $\bar{f} \in \mathcal{F}(2, 1/4) \cap \mathcal{B}(1/2)$. Furthermore, the construction (14)-(15) satisfies (13), where the perturbation term $1/d^2$ can be absorbed into the approximation error by Proposition 1:

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

In conclusion, the signal tensor constructed in the Example 3 can be regarded as a tensor generated from $\mathcal{F}(2, 5/4) \cap \mathcal{B}(1/2)$. Therefore, our Borda count algorithm is applicable to this case with the claimed accuracy in our paper. This example highlights that our marginal monotonicity assumption is weaker than the usual sense thanks to the unknown permutations.

We emphasize that, in all permutation models (Chan and Airoldi, 2014; Klopp et al., 2017; Chatterjee, 2015; Shah et al., 2019; Flammarion et al., 2019; Hütter et al., 2020),

$$\mathcal{Y} = \underbrace{\Theta \circ \pi}_{\text{signal}} + \underbrace{\mathcal{E}}_{\text{noise}} = \underbrace{(\Theta \circ \tau^{-1}) \circ (\tau \circ \pi)}_{\text{signal}} + \underbrace{\mathcal{E}}_{\text{noise}},$$

the error rate is evaluated by $\hat{\Theta} \circ \hat{\pi}$ as whole, not the individual $\hat{\Theta}$ and $\hat{\pi}$. The intrinsic non-identifiability $\Theta \circ \pi = (\Theta \circ \tau) \circ (\tau^{-1} \circ \pi)$ provides the flexibility in estimation. In your example, the

"the algorithm will mistakenly group together those top-ranked and bottom-ranked (in true ordering) nodes", yielding a wrong $\hat{\pi}$. Nevertheless, the composition $\hat{\Theta} \circ \hat{\pi}$ is still accurate, because our algorithm uses only entry values (not the entry locations) in the estimation.

– **Your specific example (practice).** We also added new simulations to verify the success of our algorithm in your example. We simulate order-3 (d, d, d)-dimensional tensors based on the latent function (12) where $d \in \{20, 30, \ldots, 100\}$. We add Gaussian noise tensor whose entries are i.i.d. drawn from $N(0, 0.5)$. Similar to other simulations, we evaluate the accuracy of the estimation by MSE and report the summary statistics across $n_{sim} = 20$ replicates. We compare **Borda Count** method with **Spectral** and **LSE** in the simulation. The hyperparameters are chosen via cross-validation that gives the best accuracy for each method. As expected, our **Borda Count** algorithm performs the best even when the true latent function is not monotonic. Our algorithm achieves the good enough error rate equivalent to $(\alpha \wedge 1)$-smoothness, outperforming other methods.

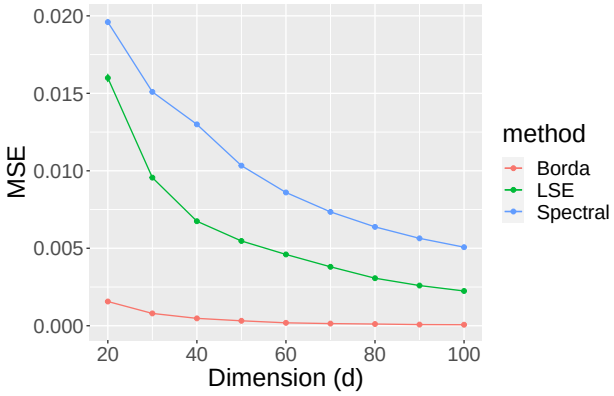


Figure R1: MSE versus the tensor dimension based on different estimation methods. The true model is defined by the latent function $f(x, y, z) = (x - 0.5)^2 + yz$.

The general setting. We realize that the exposition in the original paper may have caused the confusion. Inspired by your comments, we added the following new section in the paper:

Section 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 3 (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 4 (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.

3. *The model is 1-dimensional latent space model. All nodes are embedded into the interval of* $[0,1]$*. It is an easier case than a general latent space model where each node is embedded into a* K*-dimensional latent space. Although the block model is a special case of the current model, the current model does not include common variants of block models, such as those with mixed memberships (Jin et al., 2017). In the block model, we can encode the community labels by either a* K*-dimensional 0-1 vector* z_i *or a value* $t_i \in \{1, 2, \ldots, K\}$*. In the second representation, the latent variables are in dimension 1. However, in a model with mixed membership, only the first representation is possible. The latent variables are in dimension* K*, so finding the latent space is much more than finding an unknown permutation.*

Response: This is an excellent point! In fact, we have been considering the same extension while this paper was in submission. The extension from the block model to the mixed membership model is analogous to the extension from 1-dimensional latent space model to general dimensional latent model. This generalization extends the latent permutation $\pi \in \Pi(d,d)$ to the set of latent vectors $\boldsymbol{v}_i \in [0,1]^s$.

Our parallel work (Lee and Wang, 2023) considers the general dimensional latent variable model similar to reviewer's suggestion. The work (Lee and Wang, 2023) assumes the latent function $f\colon [0,1]^s \times \cdots \times [0,1]^s \to \mathbb{R}$ such that

$$\Theta(i_1, \ldots, i_m) = f(\boldsymbol{a}_{i_1}^{(1)}, \ldots, \boldsymbol{a}_{i_m}^{(m)}), \text{ for all } (i_1, \ldots, i_m) \in [d_1] \times \cdots \times [d_m]. \tag{16}$$

where each $a_{i_k}^{(k)}$ is an s-dimensional vector with a general $s \geq 1$. In this model, the role of unknown permutation in our paper is extended to the role of unknown latent vectors in (17). However, we find that this extension is not free. We need a stronger assumption than our α-smoothness condition for the theoretical analysis. Specifically, we need a new class called *analytic functions* such that

$$\sup_{\boldsymbol{a}_k \in \mathbb{R}^s, \|\boldsymbol{a}_k\|_\infty = 1, k \in [m]} \left| \frac{\partial^{|\kappa|} f(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m)}{\partial (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m)^\kappa} \right| \leq L^{|\kappa|} \kappa!, \quad \text{for all multi-indices } \kappa \in \mathbb{N}^m.$$

The analytic function class can be roughly viewed as ∞-smoothness plus some extra technical assumptions. The stronger assumption is the price one has to pay for more general model.

Compared to this current paper, the analysis of analytic functions uses different techniques and yields new results of its own. For this reason, we decide to not include the extension in current paper. We refer the reviewer to (Lee and Wang, 2023) if interested.

We have added the above discussion to the revised paper.

4. *I suggest that the authors do not over-claim the broadness of the current model and make a careful edit of Section 1.2. The current writing hides the "dimension-1 embedding" nature of the current model and gives readers the wrong impression that the current model is broader than the usual low-rank models.*

Response: Thank you for the suggestion. We have added the clarification in Section 1.2.

> ...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 rank might be a reasonable measure for model complexity. However, if the underlying signal is high rank but has certain shape structure, then our nonparametric approach may better capture the intrinsic model complexity....

We also have added the following paragraph in Section 7:

> 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 (Lee and Wang, 2023) 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
>
> $$\Theta(i_1, \ldots, i_m) = f(\boldsymbol{a}_{i_1}^{(1)}, \ldots, \boldsymbol{a}_{i_m}^{(m)}), \text{ for all } (i_1, \ldots, i_m) \in [d_1] \times \cdots \times [d_m]. \quad (17)$$
>
> 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 Lee and Wang (2023) for independent interest.

5. *The comparison could be unfair, if other papers did give some polynomial-time algorithms. I suggest the authors to spend more efforts reviewing the algorithms in other papers and discuss whether those algorithms are applicable to the current settings.*

Response: Thank you for the comments. Based on your comments, we added two new results.

– **In theory,** we have added a new Theorem 3 to show the *non-existence* of polynomial-time algorithm in the general setting. See **pages 1-7 of this response letter, Response to Major comments Summarized by Editor**.

– **In simulation,** we have added more algorithms for comparison. Specifically, we consider:

- Spectral method (**Spectral**) (Xu, 2018) that performs universal singular value thresholding (Chatterjee, 2015) on the unfolded tensor.
- Least-squares estimation (**LSE**) (Gao et al., 2015) which solves the optimization problem (8) with constant block approximation ($\ell = 0$) based on spectral k-means. We extend the matrix-based biclustering algorithm to higher-order tensors (Han et al., 2022).
- Least-squares estimation (**BAL**) (Balasubramanian, 2021) which solves the optimization problem (8) with constant block approximation.
- Cluster+polyℓ algorithm (suggested by the reviewer in #7) which performs clustering first and then polynomial approximation within clusters.

The results are provided in the new Figure R2 in this letter. **The comparison is summarized in our response to #7.**

6. *In the case of monotonicity, it looks very promising that the rate is faster than Pananjady and Samworth (2022), but the assumptions are different. The rate achieved by Borda count assumes infinite smoothness, which was not assumed in Pananjady and Samworth (2022). The table needs more explanations in the caption, not to mislead the readers.*

Response: Thank you for the comments. We update the table accordingly as below; the added parts are highlighted in blue.

	Pananjady and Samworth (2022)	Balasubramanian (2021)	Li et al. (2019)	**Ours** (MLE)	**Ours** (Borda count)
Model structure	monotonic	Lipschitz	Lipschitz	α-smooth	α-smooth & monotonic
Fixed grid design	$\surd$	$\times$	$\times$	$\times$	$\surd$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
(e.g., rate when $(m, \alpha) = (3, 1)$)	d^{-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 5: 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. Our method allows arbitrary smoothness level $\alpha > 0$; see Sections 3-4 for details.

7. *Clustering + blockwise polynomial approximation. In the simulations, LSE and BAL first perform clustering and then use constant block approximation. A natural question is what happens if one uses clustering followed by polynomial block approximation. Can the authors add such results into the simulations? This is also related to my Point 2. Can this approach replace the Borda count algorithm to produce a polynomial-time algorithm in the case of no monotonicity?*

Response: Thank you for the suggestion. We have added the new algorithm (denoted as cluster + polyℓ algorithm) based on the reviewer's comments. The following new result is added to the Appendix A.

> "...We consider another competing algorithm (denoted as cluster + polyℓ algorithm) for comparison. The algorithm performs clustering first and then polynomial approximation within clusters. Since we do not know the permutation, we randomly assign the order of nodes within a block. We perform the simulation using the same setting in Section 6. Figure R2 shows the comparison among all methods, including the new cluster + polyℓ algorithm for $\ell \in \{1, 2, 3\}$. Notice that cluster + poly0 algorithm is equivalent to LSE algorithm. The result shows that the cluster + polyℓ algorithms are unstable except Model 3.
>
> In fact, we are able to explain the failure of cluster + polyℓ in theory. The two original competing algorithm – constant block approximation – does not need the order of nodes. By contrast, the order information is necessary in polynomial approximation in the newly added algorithm. Specifically, the block-k degree-ℓ polynomial tensors $\mathscr{B}(k, \ell)$ is represented as
>
> $$\mathscr{B}(k,\ell) = \left\{ \mathcal{B} \in \mathbb{R}^{d\times\cdots\times d} \colon \mathcal{B}(\omega) = \sum_{\Delta\in\mathcal{E}_k} \mathrm{Poly}_{\ell,\Delta}(\omega)\mathbb{1}\{\omega \in \Delta\} \text{ for all } \omega \in [d]^m \right\},$$
>
> where $\mathrm{Poly}_{\ell,\Delta}(\cdot)$ denotes a degree-ℓ polynomial function in $\mathbb{R}^m$. In the constant block approximation, the polynomial function is $\mathrm{Poly}_{0,\Delta}(\omega) = \beta^0_\Delta$. This implies that an index

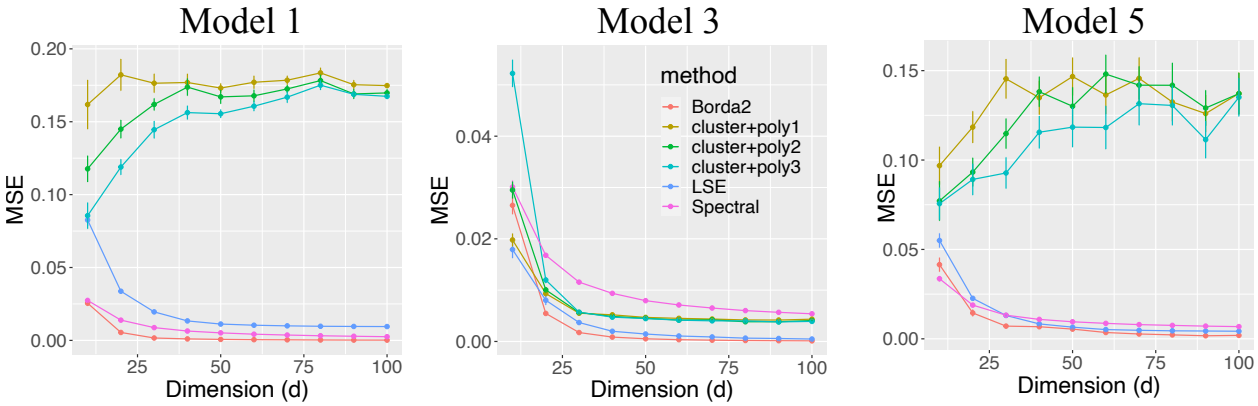


Figure R2: MSE versus the tensor dimension based on different estimation methods. Columns 1-7 consider the Models 1, 3, and 5 with continuous case in Table 2 respectively. cluster+polyℓ means polynomial ℓ-approximation

> ω does not affect the function estimation. In the linear function approximation, however, the polynomial function is $\text{Poly}_{1,\Delta}(\omega) = \langle \boldsymbol{\beta}_\Delta, \omega \rangle + \beta^0_\Delta$ for $\ell = 1$, where different index ω changes the function approximation. Therefore, polynomial approximations with degree greater than 0 require an estimate of the permutation, as in our Borda count algorithm...."

References

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

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

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

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

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

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

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

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.

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

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

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.

Hütter, J.-C., C. Mao, P. Rigollet, and E. Robeva (2020). Estimation of monge matrices. *Bernoulli 26*(4), 3051–3080.

Kiefer, J. (1982). Optimal rates for non-parametric density and regression estimation, under order restrictions. *Statistics and Probability: Essays in horner of C. R. Rao*, 419–428.

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.

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

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.

Low, M. G. and Y.-G. Kang (2002). Estimating monotone functions. *Statistical & Probability Letters 56*, 361–367.

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.

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

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

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.

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

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

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

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