Bayesian Nonlinear Tensor Regression with Functional Fused Elastic Net Prior

Shuoli Chen*, Kejun He*, Shiyuan He, Yang Ni, and Raymond K. W. Wong

Abstract

Tensor regression methods have been widely used to predict a scalar response from covariates in the form of a multiway array. In many applications, the regions of tensor covariates used for prediction are often spatially connected with unknown shapes and discontinuous jumps on the boundaries. Moreover, the relationship between the response and the tensor covariates can be nonlinear. In this article, we develop a nonlinear Bayesian tensor additive regression model to accommodate such spatial structure. A functional fused elastic net prior is proposed over the additive component functions to comprehensively model the nonlinearity and spatial smoothness, detect the discontinuous jumps, and simultaneously identify the active regions. The great flexibility and interpretability of the proposed method against the alternatives are demonstrated by a simulation study and an analysis on facial feature data.

Keywords: Additive model; sparsity; spatial smoothness; discontinuity jumps; graph Laplacian.

Authors are listed alphabetically. Shuoli Chen (Email: csljiaj@ruc.edu.cn) is Student, Kejun He (Email: kejunhe@ruc.edu.cn) is Assistant Professor, Shiyuan He (Email: heshiyuan@ruc.edu.cn) is Assistant Professor, Center for Applied Statistics, Institute of Statistics and Big Data, Renmin University of China, Beijing 100872, China. Yang Ni (Email: yni@stat.tamu.edu) is Assistant Professor, Raymond K. W. Wong (Email: raywong@tamu.edu) is Associate Professor, Department of Statistics, Texas A&M University, College Station 77843, USA.

*Corresponding authors.

1 Introduction

Data in the form of multiway arrays, also known as tensors, are becoming increasingly common in physical and engineering sciences. For example, Yan et al. (2019) studied the machinability of titanium alloy where the cylinder-shaped materials are represented by multidimensional arrays. Yue et al. (2020) performed quality inspections of nanomanufacturing processes with Raman spectral imaging data which are formulated as a tensor. Zhong et al. (2022) proposed a tensor-based approach to handle the spatial and temporal structures of image outputs in the automatic control processes of semiconductor manufacturing. In hot rolling processes, multiple sensors record the temperature, current, torque, speed at an equal time interval, generating multiple signals in form of tensors (Miao et al., 2021). Shi (2023) provided a good review for some recent applications of statistical tensor methods in manufacturing quality improvement. Tensor data are also important in many other areas such as chemometrics (Andersen and Bro, 2003), text mining (Chew et al., 2007), and recommendation systems (Park and Chu, 2009). Among the successful applications of tensor data analysis, using tensor regression to decode the relationship between a scalar response and the covariates of a tensor structure has attracted considerable attentions. In condition monitoring and industrial asset management, Fang et al. (2019) applied a tensor regression model to predict the residual lifetime of a rotating machinery according to the degradation image streams acquired using an infrared camera. In neuroscience, researchers apply tensor regression methods to predict diseases and disorders such as Alzheimer's disease (Kandel et al., 2013) and autism spectrum disorder (Ecker et al., 2013) based on the magnetic resonance imaging or diffusion tensor imaging of human brain.

A general scalar-on-tensor regression model between a D-way tensor of covariates $\mathbf{X} \in \mathbb{R}^{P_1 \times \cdots \times P_D}$ and a response $Y \in \mathbb{R}$ can be formulated via a regression function $f : \mathbb{R}^{P_1 \times \cdots \times P_D} \to \mathbb{R}$ and an additive noise: $Y = f(\mathbf{X}) + \epsilon$. The majority of existing tensor regression methods adopts the linear regression form $f(\mathbf{X}) = \sum_{i_1,\cdots,i_D} X_{i_1,\cdots,i_D} \beta_{i_1,\cdots,i_D}$ where $\beta_{i_1,\cdots,i_D}$ is the $(i_1, \cdots, i_D)$-th element of the tensor coefficient $\boldsymbol{\beta} \in R^{P_1 \times \cdots \times P_D}$ to be estimated. To overcome the difficulty of estimating a huge number of coefficients in many tensor applications, Zhou et al. (2013) proposed a linear tensor regression model with a low-rank structure of $\boldsymbol{\beta}$ via the

CANDECOMP/PARAFAC (CP) decomposition (Harshman, 1970). Additional regularization methods such as the lasso (Tibshirani, 1996) and the ridge (Hoerl and Kennard, 1970) were also suggested to obtain a consistent and interpretable estimator. Guhaniyogi et al. (2017) proposed Bayesian Tensor Regression (BTR), which again utilized the CP decomposition. With carefully constructed shrinkage priors, BTR is able to shrink parameters at both local and global levels, and select the rank automatically. Some other works of tensor linear regression are based on different types of decomposition including Tucker decomposition (Tucker, 1966) on the coefficient tensor $\boldsymbol{\beta}$ (Li et al., 2018).

However, the assumption that the tensor covariates can predict the response through a linear regression function is too restrictive and can be violated in many applications. For instance, in the field of financial analysis, Li et al. (2016) found that the nonlinearity exists in the relationship between stock movements and information sources in the form of tensor data. To model the nonlinearity of regression function f while keeping the inherent structural information of the original tensor, Zhao et al. (2013, 2014) placed a Gaussian process prior over the regression function where the covariance function is a product kernel based on the unfoldings of tensor covariates. With a rank-1 CP decomposition $\mathbf{X} = \mathbf{x}_1 \circ \cdots \circ \mathbf{x}_D$ where $\circ$ denotes the outer product and $\mathbf{x}_d$ is a P_d-dimensional vector, Signoretto et al. (2013) and Kanagawa et al. (2016) considered a regression model $f(\mathbf{X}) = \sum_{r=1}^{R} \prod_{d=1}^{D} f_r^{(d)}(\mathbf{x}_d)$ with a Gaussian process prior over each $f_r^{(d)}$, $d = 1, \ldots, D$. Extending the rank-1 assumption, a more flexible model $f(\mathbf{X}) = \sum_{r=1}^{R} \sum_{m=1}^{M} \prod_{d=1}^{D} f_r^{(d)}(\mathbf{x}_d^{(m)})$ with $\mathbf{X} = \sum_{m=1}^{M} \mathbf{x}_1^{(m)} \circ \cdots \circ \mathbf{x}_D^{(m)}$ was proposed in Imaizumi and Hayashi (2016). Unfortunately, a number of multidimensional functions have to be estimated in the above work, which will suffer from the curse of dimensionality when some P_d's are large. An alternative approach of modeling the nonlinear regression function is using the similar idea of additive models (Stone, 1985) on the vector of covariates. Nonparametric additive models have recently been extended to tensor covariates with elastic net (Zhou et al., 2020) and the group lasso penalty (Hao et al., 2021). They again exploit the tensor structure through CP decomposition of the tensor coefficient.

In many applications, the tensor of covariates (e.g., a 3D image) is a collection of observations at a regular grid over a multidimensional continuous domain. One common observation in the corresponding applications is the existence of spatially contiguous active regions with

unknown shapes and discontinuous jumps on the boundaries of regions, especially in image data. For example, in neuroscience, the pathological studies show that the brain voxels that have significant effects to the diseases are expected to be sparse and organized into several spatially connected regions (Michel et al., 2011; Fiot et al., 2014). Therefore, the presence of multiple piecewise smooth regions should be considered in the regression function f. Although there exist prior works that are related to the modeling of this spatial structure, such as Xin et al. (2014); Goldsmith et al. (2014); Li et al. (2015); Wang et al. (2017); Beer et al. (2019), most make the linear assumption on the regression function. One notable exception is Marx et al. (2011), which proposed a nonlinear tensor regression with spatial similarity through a single-index model. However, their method does not produce sparse estimation, and thus the important subregions are hard to be identified using their model. In this work, we propose a novel Bayesian tensor additive regression model that incorporates the spatial structure of tensor covariates and strikes a good balance between flexibility and interpretability. More precisely, we design a prior called functional fused elastic net (FEN) over the nonlinear additive component functions to adaptively learn the spatial smoothness of the component functions within unknown connected regions. The spatial smoothness is achieved by the graph Laplacian of the adjacent entries, and discontinuous jumps between distinct regions are detected by the ℓ_1 fusion of the adjacent entries. With spline representation, we apply the idea of the thresholding method (Ni et al., 2019; Cai et al., 2020) on the coefficients to achieve sparsity and identify the important regions. A crucial advantage of thresholding method against the common alternatives, such as spike-and-slab priors (Mitchell and Beauchamp, 1988) and Bayesian credible intervals (Chen and Shao, 1999), is its low computation cost and the ability to drop the inactive signals without increasing the predictive error. The posterior inference is carried out through a Markov chain Monte Carlo (MCMC) method with the Metropolis-adjusted Langevin Algorithm (MALA, Roberts and Rosenthal, 1998). To the best of our knowledge, our work is the first to integrate the spatial smoothness and discontinuous jumps for sparse nonlinear tensor regression.

The rest of this paper is organized as follows. In Section 2, we present the tensor additive model and introduce the spatially piecewise smooth structure to integrate the idea of sparsity, spatial smoothness, and discontinuous jumps. Section 3 proposes the functional FEN prior

for the component functions of the tensor additive model and illustrates its properties with some examples. Using spline expansion to approximate each additive component function, a Bayesian hierarchical model is formulated on the spline coefficients, and a posterior sampling algorithm is described. A simulation study and a real application on facial feature data are respectively presented in Sections 4 and 5 to demonstrate the advantages of the proposed model over existing alternatives. We finally summarize this article in Section 6 with some concluding remarks.

2 Tensor Additive Regression Model

We consider the scalar-on-tensor regression setting where the covariate $\mathbf{X} \in \mathbb{R}^{P_1 \times \cdots \times P_D}$ is a D-way tensor of dimension $P_1 \times \cdots \times P_D$ and the response $Y \in \mathbb{R}$ is a scalar. The element $X_{\mathbf{i}}$ of $\mathbf{X}$ is indexed by $\mathbf{i} \in \mathcal{I} = \{(i_1, i_2, \cdots, i_D) : 1 \leq i_d \leq P_d, 1 \leq d \leq D\}$. Without loss of generality, we assume $X_{\mathbf{i}} \in [0, 1]$ for all $\mathbf{i}$. The number of elements in $\mathbf{X}$ can be much larger than the sample size in many applications. For example, the magnetic resonance imaging dataset considered in Zhou et al. (2013) consists of 776 patients with the number of covariates up to $256 \times 198 \times 256 = 12,976,128$. High dimensionality leads to significant difficulties in modeling the nonlinear regression function. A natural nonlinear regression model is a tensor additive model:

$$Y = \mu + \sum_{\mathbf{i} \in \mathcal{I}} f_{\mathbf{i}}(X_{\mathbf{i}}) + \epsilon, \quad \epsilon \sim N(0, \sigma_\epsilon^2), \tag{1}$$

where $f_{\mathbf{i}}$'s are nonlinear functions such that $\int_0^1 f_{\mathbf{i}}(x)\,\mathrm{d}x = 0$ for all $\mathbf{i}$ (for identifiability purposes). However, even with the additive model assumption, there are still a potentially huge number of univariate nonparametric functions to be estimated. With a limited amount of data, it is often challenging to estimate these functions well. Furthermore, there are three types of useful structures in tensor regressions, which are not incorporated by model (1).

Sparsity. In many real applications, only a few entries of the tensor covariates may be relevant to predict the response. Take neuroimaging as an example, the brain is believed to have dedicated regions for different tasks. For instances, the visual cortex in human brains controls visual functions (Grill-Spector and Malach, 2004) and the frontal lobe is responsible for reasoning (Collins and Koechlin, 2012). We thus generally expect many

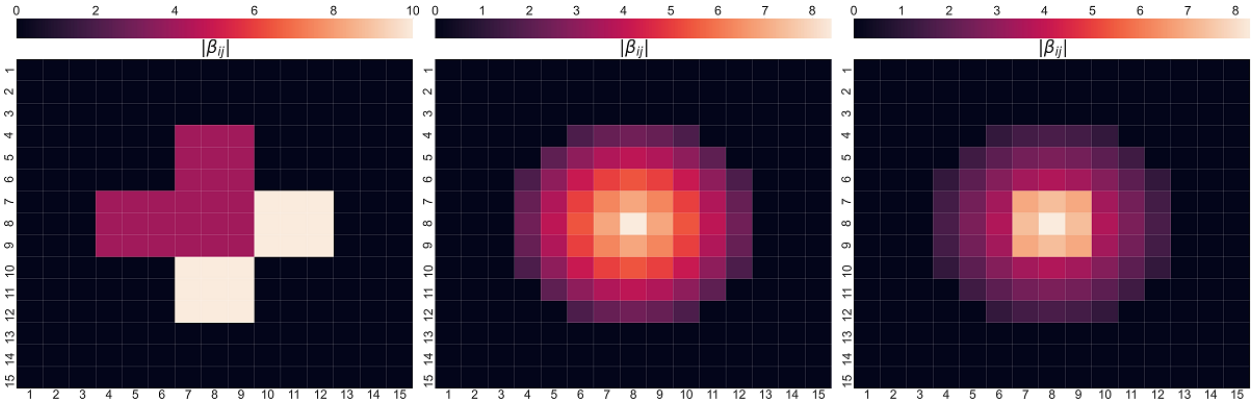

Figure 1: Three examples of 2-way tensor additive model with spatially piecewise smooth structure. The heatmaps show the magnitude of each additive component.

additive component functions in (1) to be zero (i.e., $f_{\mathbf{i}} \equiv 0$) for predicting reasoning and visual-related outcomes. In the following, the sets of $\mathbf{i}$ where the additive component function $f_{\mathbf{i}}$ is non-zero and zero are called active regions and non-active regions, respectively.

Spatial smoothness. We further assume the additive model (1) to be endowed with a *spatially smooth functional structure*, which means that the functions $f_{\mathbf{i}}$'s vary smoothly with respect to the location index $\mathbf{i}$. Specifically, the functions $f_{\mathbf{i}}$'s are spatially smooth with respect to a graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where $\mathcal{E}$ is the neighboring relationship set for the location index set $\mathcal{I}$. A pair of indices $(\mathbf{i}, \mathbf{i}') \in \mathcal{E}$ are connected by an edge when $X_{\mathbf{i}}$ and $X_{\mathbf{i}'}$ are neighboring elements in the tensor of covariates $\mathbf{X}$. Equivalently, $(\mathbf{i}, \mathbf{i}') \in \mathcal{E}$ if $\|\mathbf{i} - \mathbf{i}'\|_1 = 1$, where $\|\cdot\|_1$ represents the ℓ_1-norm. For the additive model (1) to be spatially smooth with respect to $\mathcal{G}$, functions $f_{\mathbf{i}}, f_{\mathbf{i}'}$ with $(\mathbf{i}, \mathbf{i}') \in \mathcal{E}$ are likely to be similar to each other.

Discontinuous jumps. Sparsity and spatial smoothness together require the functions to be smoothly decaying to zero towards the boundary of an active/non-zero region. This may not be realistic. In a natural image or neuroimage, a pixel (or voxel) at the boundary of an active region could have significant effect on the response. Our work aims to address this challenging issue, by developing a spatially smooth model that allows for occasional discontinuous jumps, i.e., if supported by data, a few $f_{\mathbf{i}}$'s can vary non-smoothly from its neighbors.

Combining spatial smoothness and discontinuity, we obtain a hybrid structure, which we call *spatially piecewise smooth functional structure*. More specifically, in this structure, the

index set $\mathcal{I}$ can be divided into a few distinct spatially connected regions $\mathcal{I}_1, \cdots, \mathcal{I}_C$, and the component functions within the same region are spatially smooth. Discontinuity are allowed on the boundary between regions.

Figure 1 illustrates the various types of spatially piecewise smooth functional structures that our model can handle. It shows the heatmap of $|\beta_{\mathbf{i}}|$ for function $f_{\mathbf{i}}(X_{\mathbf{i}}) = \beta_{\mathbf{i}} X_{\mathbf{i}}$, which is linear for the simplicity of illustration. In the left panel, the active (non-black) regions can be divided into three pieces. Inside each piece, $|\beta_{\mathbf{i}}|$ is spatially smooth (in fact, it is a constant). There are discontinuity jumps between the active and non-active regions and between each pair of active regions. The middle panel simply contains one active region and is overall smooth. The right panel has a discontinuity jump at the central square, and is spatially smooth within the central square and the surrounding circle, respectively.

3 Bayesian Model

In this section, we develop a Bayesian hierarchical model for the inference of the tensor additive model (1). We propose a *functional fused elastic net (functional FEN)* prior to deal with the spatially piecewise smooth functional structure and illustrate its advantage through two simple numerical experiments. Using basis representation, we show that the proposed functional FEN prior can be transferred to a proper prior on the the basis coefficients. An efficient computational algorithm for the posterior inference is also developed.

3.1 Functional Fused Elastic Net Prior

To construct a prior distribution that encourages sparsity, each $f_{\mathbf{i}}$ is parameterized as the product of a latent function $g_{\mathbf{i}} \in C^2[0,1]$ and a hard thresholding function $\mathbf{1}_{\{\|g_{\mathbf{i}}\|^2_{\mathbb{L}_2} > \lambda\}}$, i.e.,

$$f_{\mathbf{i}} = g_{\mathbf{i}} \cdot \mathbf{1}_{\{\|g_{\mathbf{i}}\|^2_{\mathbb{L}_2} > \lambda\}}, \quad \mathbf{i} \in \mathcal{I}, \tag{2}$$

where λ is the thresholding parameter. Roughly speaking, $f_{\mathbf{i}}$ is thresholded to exact zero $f_{\mathbf{i}} \equiv 0$ whenever the latent function $g_{\mathbf{i}}$ has a small magnitude. Using the form of (2), the spatially piecewise smooth functional structure on $f_{\mathbf{i}}$ can be equivalently modeled on $g_{\mathbf{i}}$.

Let $\ell[0,1]$ denote the set of affine functions on the interval $[0,1]$, i.e.,

$$\ell[0,1] = \{l(x) : l(x) = a + bx\}. \tag{3}$$

Denote the projection operator from the space of the second order Sobolev space $\mathcal{W}_2^2[0,1]$ onto $\ell[0,1]$ by $\mathcal{P}$. We propose a functional FEN prior distribution for the set of all the latent functions $\boldsymbol{G} = \{g_{\mathbf{i}}(x) : \mathbf{i} \in \mathcal{I}\}$:

$$p\big(\boldsymbol{G}|\,\delta, r_1, r_2\big) \propto \exp\Big\{ -\delta \sum_{\mathbf{i}\in\mathcal{I}} \mathcal{R}(g_{\mathbf{i}}) - r_1 \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|g_{\mathbf{i}} - g_{\mathbf{i}'}\|_{\mathbb{L}_2} - r_2 \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|g_{\mathbf{i}} - g_{\mathbf{i}'}\|_{\mathbb{L}_2}^2 \Big\}, \tag{4}$$

where $\mathcal{R}(g_{\mathbf{i}}) = \|g_{\mathbf{i}}''\|_{\mathbb{L}_2}^2 + \delta' \|\mathcal{P} g_{\mathbf{i}}\|_{\mathbb{L}_2}^2$ measures the roughness of $g_{\mathbf{i}}$ with $g_{\mathbf{i}}''$ being the second derivative of $g_{\mathbf{i}}$ and $\delta' \in \mathbb{R}^+$. The second summation in the prior distribution (4) is the functional *fusion* term, which encourages local constant structure and helps build the piecewise structure. The third summation is the functional *Laplacian* term, which encourages spatial smoothness.

The fusion and the Laplacian terms of the functional FEN prior distribution (4) can be viewed as an *adaptive* Laplacian prior distribution. To see this, we use a Gaussian scale mixture identity as follows:

$$e^{-b|x|} = \int \frac{1}{\sqrt{\pi\omega}} \exp\left(\frac{b^2 x^2}{4\omega}\right) \cdot e^{-\omega}\, \mathrm{d}\omega.$$

We can rewrite the fusion term in (4) through independent latent random variables $\omega_{\mathbf{i}\mathbf{i}'}$, $(\mathbf{i},\mathbf{i}') \in \mathcal{E}$, following the standard exponential distribution,

$$p\big(\boldsymbol{G}|\,\delta, r_1, r_2, \omega_{\mathbf{i}\mathbf{i}'}\big) \propto \prod_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \frac{1}{\sqrt{\omega_{\mathbf{i}\mathbf{i}'}}} \exp\Big\{ -\delta \sum_{\mathbf{i}\in\mathcal{I}} \mathcal{R}(g_{\mathbf{i}}) - \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \big(r_2 + \frac{r_1^2}{4\omega_{\mathbf{i}\mathbf{i}'}}\big) \|g_{\mathbf{i}} - g_{\mathbf{i}'}\|_{\mathbb{L}_2}^2 \Big\},$$
$$\omega_{\mathbf{i}\mathbf{i}'} \overset{\text{i.i.d.}}{\sim} \mathrm{Exp}(1) \text{ for all } (\mathbf{i},\mathbf{i}') \in \mathcal{E}.$$

Using this representation, the second and the third summation in (4) are merged into a single term. The prior distribution generally encourages the neighboring functions to be similar, i.e., with small $\mathbb{L}_2$ distance. When $\omega_{\mathbf{i}\mathbf{i}'}$ is close to zero, its contribution to the prior distribution could be very large. Thus, for the corresponding neighboring functions $g_{\mathbf{i}}$ and $g_{\mathbf{i}'}$, the prior has the tendency to push them towards being identical. In the next subsection, we discuss more properties of the functional FEN prior and show how the fusion and the Laplacian terms successfully accommodate a spatially piecewise smooth structure.

3.2 Properties of the Fuison and Laplacian Prior

For the proposed functional FEN prior distribution (4), both the functional fusion term and the functional Laplacian term play indispensable roles. For simplicity, we illustrate their importance via a special setting where each additive component function is linear with $f_{\mathbf{i}}(X_{\mathbf{i}}) = X_{\mathbf{i}}\beta_{\mathbf{i}}$ and $\beta_{\mathbf{i}} \in \mathbb{R}$, $\mathbf{i} \in \mathcal{I}$. In this setting, the functional FEN prior (4) reduces to a prior on the scalars $\beta_{\mathbf{i}}$'s as

$$p(\boldsymbol{\beta}|\delta, r_1, r_2) \propto \exp\Big\{ -\delta \sum_{\mathbf{i}\in\mathcal{I}} \beta_{\mathbf{i}}^2 - r_1 \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} |\beta_{\mathbf{i}} - \beta_{\mathbf{i}'}| - r_2 \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} (\beta_{\mathbf{i}} - \beta_{\mathbf{i}'})^2 \Big\}. \tag{5}$$

From (5), we observe that when $\delta = r_2 = 0$, the corresponding prior of $p(\boldsymbol{\beta}|0, r_1, 0)$ degenerates to the generalized fused lasso (Tibshirani et al., 2005). When $\delta = r_1 = 0$, the FEN prior reduces to Laplacian prior or Gaussian Markov random field (Rue and Held, 2005). When $\delta = 0$, the corresponding (negative log) FEN prior, i.e., $-\log p(\boldsymbol{\beta}|0, r_1, r_2)$, is equivalent to the graph-fused elastic net penalty (Tec et al., 2019).

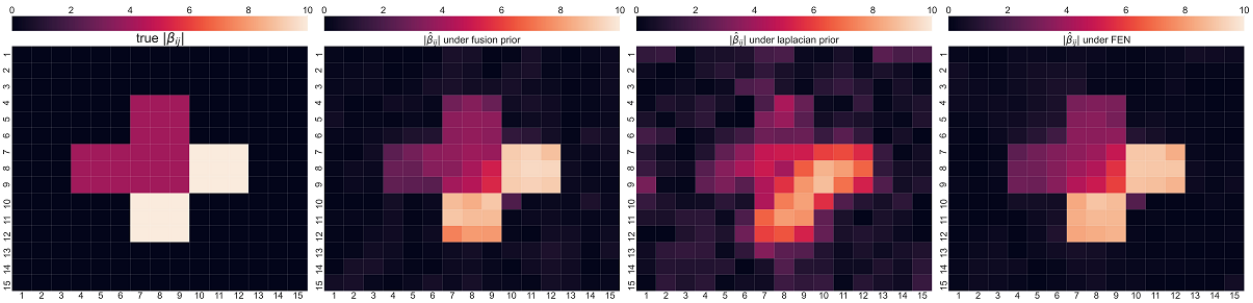

Figure 2: A toy simulation where the component functions are linear, i.e. $f_{\mathbf{i}}(X_{\mathbf{i}}) = X_{\mathbf{i}}\beta_{\mathbf{i}}$. From left to right, the four panels correspond to the true values of $\beta_{\mathbf{i}}$, the posterior mean of $\beta_{\mathbf{i}}$'s with the fusion prior, the Laplacian prior and FEN prior, respectively.

We conduct two simple experiments to illustrate the properties of fusion prior and Laplacian prior, and show the performance gain of FEN by combining them. For these two experiments, we set $\mathcal{I} = \{(i,j) : 1 \le i,j \le 15\}$, and the matrix covariates are generated by $X_{ij} \overset{\text{i.i.d.}}{\sim} \text{Unif}(0,1)$. The responses are generated according to $y = \sum_{1\le i,j\le 15} X_{ij}\beta_{ij} + \epsilon$ with $\epsilon \sim N(0,1)$, and the true values of β_{ij}'s are shown in the leftmost panels of Figures 2 and 3, respectively for two experiments. These settings correspondingly feature the true model with the following structures: 1) a spatially piecewise constant structure, and 2) a spatially

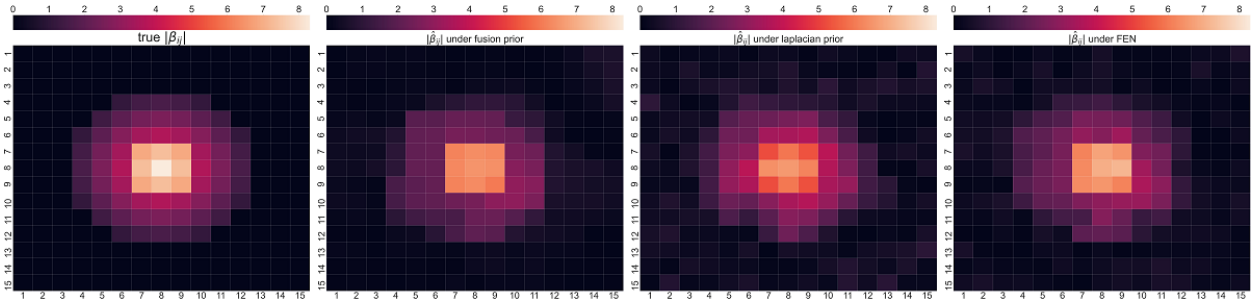

Figure 3: A toy simulation where the component functions are linear, i.e. $f_{\mathbf{i}}(X_{\mathbf{i}}) = X_{\mathbf{i}}\beta_{\mathbf{i}}$. From left to right, the four panels correspond to the true values of $\beta_{\mathbf{i}}$, the posterior mean of $\beta_{\mathbf{i}}$'s with the fusion prior, the Laplacian prior and FEN prior, respectively.

piecewise smooth structure. We generate $N = 100$ observations for each setting and repeat the experiments for 30 times. The posterior distribution of $\boldsymbol{\beta}$ is given by

$$p(\boldsymbol{\beta}|D_N, \delta, r_1, r_2) \propto \exp\Big\{ -\frac{1}{2}\sum_{n=1}^{N}(y_n - \langle \boldsymbol{\beta}, \mathbf{X}^{(n)}\rangle)^2 \Big\} \times p(\boldsymbol{\beta}|\delta, r_1, r_2), \tag{6}$$

where the prior distribution $p(\boldsymbol{\beta}|\delta, r_1, r_2)$ is given in (5). For simplicity, we fix $\delta = 0$ and vary the hyperparameters r_1, r_2 to achieve the fusion prior ($r_2 = 0$), the Laplacian prior ($r_1 = 0$), and the general FEN prior. For the FEN prior we adopt parameterization $r_1 = r/\rho$, $r_2 = (1-r)/\rho$ with candidate grids $r \in \{1, 0.75, 0.5, 0.25, 0\}$ and $\rho \in \{0.3, 0.6, 1.2, 2.4, 4.8\}$. MALA (Roberts and Rosenthal, 1998) is applied to draw posterior samples from the model. The hyperparameters are selected as those with best predictive performance on a validation dataset. We randomly pick one replication from each experiment setting and show the posterior mean of $\boldsymbol{\beta}$ from the fusion, Laplacian, FEN priors in the second, third, and fourth panels of Figures 2 and 3. The performances of methods are also evaluated in terms of MSE $= \frac{1}{15\times 15}\sum_{i,j}(\beta_{ij} - \widehat{\beta}_{ij})^2$ where $\widehat{\beta}_{ij}$ is the posterior mean of β_{ij}. The average MSE over 30 replicates are summarized in Table 1.

Figure 2 and Table 1 reveal that, when the true model has a spatially piecewise constant structure, the fusion prior ($r_2 = 0$) has a smaller MSE than the Laplacian prior ($r_1 = 0$). The estimated β_{ij}'s from the fusion prior (the second panel) recover the true signal pattern reasonably well. However, the true pattern has been smoothed out by the Laplacian prior (the third panel). The FEN prior selects $r_2 = 0$ in all 30 replicates, and hence its performance (the fourth panel) is similar to that of the fusion prior.

Table 1: The performance of fusion, Laplacian and FEN priors under different true models for 30 random replicates. The numbers in the parentheses are the standard errors.

True Model	spatially piecewise constant			spatially piecewise smooth		
Prior	fusion	Laplacian	FEN	fusion	Laplacian	FEN
MSE	0.4286 (0.0204)	2.4860 (0.0635)	0.4317 (0.0189)	0.1707 (0.0080)	0.3265 (0.0075)	0.1589 (0.0059)

The above results demonstrate the advantage of fusion prior over Laplacian prior in estimating spatially piecewise constant model, which is consistent with the findings in Tibshirani et al. (2005) and Little and Jones (2010). However, the fusion prior tends to force similar neighboring values to be identical, and so it may introduce biases when the true values are not exactly constant. Figure 3 shows an example the Laplacian prior and the fusion prior can be combined to tackle more challenging settings. As shown in the leftmost panel, the true model is spatially piecewise smooth. There are discontinuity jumps on the boundary between a center square piece and a surrounding circle piece, and the signals vary smoothly within each piece. Neither the fusion prior nor the Laplacian prior estimates β_{ij}'s accurately in this case. The fusion prior over-shrinks the coefficient in the center square piece and the surrounding circle piece to a constant, while the Laplacian prior over-smooths the estimates globally. By contrast, the FEN prior, which combines the fusion and the Laplacian priors, is able to capture the corresponding piecewise smooth structure and has the lowest MSE.

3.3 Spline Representation of Functions

To facilitate the estimation of the unknown functions, we expand $g_{\mathbf{i}}(x) = \sum_{k=1}^{K} \alpha_{\mathbf{i}k}\phi_k(x)$ via a vector of spline basis functions $\boldsymbol{\phi}(x) = (\phi_1(x), \ldots, \phi_K(x))^{\mathrm{T}}$, where $\boldsymbol{\alpha}_{\mathbf{i}} = (\alpha_{\mathbf{i}1}, \ldots, \alpha_{\mathbf{i}K})^{\mathrm{T}}$ is the vector of spline coefficients, $\mathbf{i} \in \mathcal{I}$. Denote $\boldsymbol{\alpha}^{\mathrm{T}} = (\boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}})_{\mathbf{i}\in\mathcal{I}} \in \mathbb{R}^{P_1\times\cdots\times P_D\times K}$. We require the vector of basis functions $\boldsymbol{\phi}(\cdot)$ to have the following properties.

(i) The basis functions are centered, i.e., $\int \boldsymbol{\phi}(x)\,\mathrm{d}x = \mathbf{0}$. This guarantees $\int g_{\mathbf{i}}(x)\,\mathrm{d}x = \int \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\boldsymbol{\phi}(x)\,\mathrm{d}x = 0$ for any $\boldsymbol{\alpha}_{\mathbf{i}} \in \mathbb{R}^K$ and thus $\int f_{\mathbf{i}}(x)\,\mathrm{d}x = 0$ due to (2).

(ii) The basis functions are orthonormal, i.e., $\int \boldsymbol{\phi}(x)\boldsymbol{\phi}(x)^{\mathrm{T}}\,\mathrm{d}x = \mathbf{I}_K$. As such, the $\mathbb{L}_2$ norm of function $g_{\mathbf{i}}$ can be directly evaluated as the Euclidean norm of the spline coefficients, $\|g_{\mathbf{i}}\|_{\mathbb{L}_2}^2 = \|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2$. This also facilitates the representation of (2) as $f_{\mathbf{i}}(X_{\mathbf{i}}) = \boldsymbol{\phi}(X_{\mathbf{i}})^{\mathrm{T}}\boldsymbol{\beta}_{\mathbf{i}}$

with

$$\boldsymbol{\beta}_{\mathbf{i}} = \boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{1}_{\{\|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 > \lambda\}}. \tag{7}$$

(iii) The second derivatives of the basis functions are orthogonal, i.e., $\boldsymbol{\Omega} := \int \boldsymbol{\phi}''(x)\boldsymbol{\phi}''(x)^{\mathrm{T}}\,\mathrm{d}x = \mathrm{diag}(\omega_{11}, \omega_{22}, \cdots, \omega_{KK})$. The $\mathbb{L}_2$ norm of $g_{\mathbf{i}}''$ can thus be directly evaluated by the weighted Euclidean norm of the spline coefficients, i.e., $\|g_{\mathbf{i}}''\|_{\mathbb{L}_2}^2 = \sum_{k=1}^{K} \omega_{kk}\alpha_{\mathbf{i}k}^2$.

In the above, the first property is for the identifiability of the additive model (1). The second and the third reduce the complexity of calculating $\mathbb{L}_2$ norm of $g_{\mathbf{i}}$ and $g_{\mathbf{i}}''$ from $\mathcal{O}(K^2)$ to $\mathcal{O}(K)$. A vector of basis functions $\boldsymbol{\phi}$ that satisfies these conditions can be constructed from B-spline basis functions, and the details are provided in Section A.1 of the Appendix. We also show that the first basis function, ϕ_1, in the constructed bases satisfies $\phi_1 \in \ell[0,1]$ as defined in (3).

Using the basis functions $\boldsymbol{\phi}$, the functional FEN prior (4) can be written as

$$p(\boldsymbol{\alpha}|\delta, r_1, r_2) \propto \exp\Big(-\delta \sum_{\mathbf{i}\in\mathcal{I}} \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\mathbf{R}\boldsymbol{\alpha}_{\mathbf{i}} - r_1 \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2 - r_2 \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2 \Big). \tag{8}$$

In (8), $\mathbf{R} = \boldsymbol{\Omega} + \delta'\|\phi_1\|_{\mathbb{L}_2}^2 \mathbf{e}_1\mathbf{e}_1^{\mathrm{T}}$, where $\mathbf{e}_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$.

3.4 The Hierarchical Bayesian Model

We now summarize our hierarchical model. Given the intercept μ, the spline coefficients $\boldsymbol{\alpha}$, the residual variance σ^2, and the thresholding parameter λ, the response y_n for the n-th observation follows a Gaussian distribution,

$$y_n|\mu, \boldsymbol{\alpha}, \mu, \sigma^2, \lambda \overset{\text{i.i.d.}}{\sim} N\Big(\mu + \sum_{\mathbf{i}\in\mathcal{I}} \phi(X_{\mathbf{i}}^{(n)})^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{1}_{\{\|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 > \lambda\}}, \sigma^2\Big). \tag{9}$$

A weakly informative Gaussian prior is imposed for μ, and a generalized inverse Gaussian distribution $\mathrm{GIG}(p, a, b)$ is imposed for the thresholding parameter λ, i.e.,

$$\mu \sim N(0, \sigma_\mu^2) \quad \text{and} \quad \ln p(\lambda) \propto (p-1)\ln\lambda - \frac{a/\lambda + b\lambda}{2}. \tag{10}$$

The generalized inverse Gaussian distribution keeps λ away from 0 and meanwhile prevents λ from being too large. The prior (8) of the spline coefficients $\boldsymbol{\alpha}$ is re-parameterized as

$$p(\boldsymbol{\alpha} \mid \delta, r, \sigma^2, \rho_{\boldsymbol{\alpha}}) = \frac{1}{C_{\delta,\sigma^2}} \cdot \exp\Big(- \frac{\delta \sum_{\mathbf{i}\in\mathcal{I}} \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}} \mathbf{R} \boldsymbol{\alpha}_{\mathbf{i}}}{\sigma^2} - \frac{r \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2 + (1-r) \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2}{2\sigma^2 \rho_{\boldsymbol{\alpha}}} \Big), \quad (11)$$

where C_{δ,σ^2} is a normalizing term, δ and $\rho_{\boldsymbol{\alpha}}$ control the informativeness of the prior, and r controls the relative weights of the Laplacian prior and fusion prior. The normalizing term C_{δ,σ^2} depends on δ and σ^2 and is not analytically available. Therefore, to facilitate computation, we propose a joint prior for σ^2 and δ,

$$p(\delta, \sigma^2) \propto C_{\delta,\sigma^2} \cdot \delta^{p_0 - 1} \exp(-\delta) \cdot \Big(\frac{1}{\sigma^2}\Big)^{p_1+1} \exp\Big(- \frac{1}{\sigma^2}\Big), \quad (12)$$

which includes the normalizing term C_{δ,σ^2} in (11). This construction allows the normalizing term to be canceled out when deriving the full conditional of δ and σ^2.

However, special care is needed to ensure that (12) is proper and also weakly informative. For propriety, the integral of (12) is finite if and only if the integral with respect to δ in the neighborhood of 0 and the integral with respect to σ^2 in the neighborhood of $+\infty$ are both finite. Hence we need to derive the order of magnitude of C_{δ,σ^2} as $\sigma^2 \to \infty$ and $\delta \to 0$. Proposition 1 below shows p_0 and p_1 in (12) should be at least larger than $K/2$ and $P_1 \cdots P_D K/2$, respectively.

Proposition 1. *The order of magnitude of the normalizing term C_{δ,σ^2} satisfies: (i) with respect to δ, C_{δ,σ^2} is of order $(1/\delta)^{K/2}$ as $\delta \to 0$, and of order $(1/\delta)^{P_1 \cdots P_D K/2}$ as $\delta \to +\infty$; (ii) with respect to σ^2, C_{δ,σ^2} is of order $(\sigma^2)^{(2P_1 \cdots P_D - 1)K/2}$ as $\sigma^2 \to 0$, and of order $(\sigma^2)^{P_1 \cdots P_D K/2}$ as $\sigma^2 \to +\infty$.*

The proof is provided in Section A.2 of the Appendix. For weak informativeness, we suggest to standardize the response variable so that the variance σ^2 of noise ϵ should concentrate on $[0, 1]$. Because Proposition 1 shows that C_{δ,σ^2} is of order $\big(\sigma^2\big)^{(2P_1 \cdots P_D - 1)K/2}$ as $\sigma^2 \to 0$, we set $p_1 = (2P_1 \cdots P_D - 1)K/2$ to balance the magnitude of C_{δ,σ^2} and thus make (12) weakly informative with respect to σ^2 (similar to an inverse-gamma prior with the shape parameter close to 0 near the origin). As for the hyperparameter p_0, it is associated with the parameter

δ, which controls the smoothness of the function. We will determine p_0 in a data-adaptive way and present the details in Section A.5.3 of the Appendix.

Although the approximation of additive component functions $f_{\mathbf{i}}$'s and the definition of $\mathbf{R}$ in prior (11) are based on the vector of spline basis functions $\boldsymbol{\phi}$, the posterior distribution of $\sum_{\mathbf{i}\in\mathcal{I}} f_{\mathbf{i}}(x)$ remains unchanged for the proposed hierarchical model if an equivalent vector of orthonormal bases of the same spline space is employed. This invariant property of our hierarchical model is summarized in Proposition 2 and its proof is presented in Section A.3 of the Appendix.

Proposition 2. *The inference for tensor additive regression (1) is invariant with respect to an orthonormal transformation of the basis functions. That is, for $\boldsymbol{\phi}_{\mathbf{Q}} = \mathbf{Q}\boldsymbol{\phi}$ where $\mathbf{Q} \in \mathbb{R}^{K\times K}$ is orthonormal, the posterior distribution of $f := \sum_{\mathbf{i}\in\mathcal{I}} f_{\mathbf{i}}$ remains unchanged.*

3.5 Posterior Sampling Algorithm

We apply a hybrid MCMC method to obtain the posterior samples of $\{\mu, \boldsymbol{\alpha}, \lambda, \sigma^2, \delta\}$ from the hierarchical model (9)–(12). In particular, the MALA (Roberts and Rosenthal, 1998) is used to sample μ and $\boldsymbol{\alpha}$; the parameters σ^2 and δ are drawn from their full conditional probabilities; and the Metropolis-Hastings algorithm with a truncated normal proposal is applied to update λ (Cai et al., 2020). As MALA requires the posterior to be differentiable, we approximate the non-differentiable components of the posterior by

$$\mathbf{1}_{\{\|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2>\lambda\}} \approx t(\boldsymbol{\alpha}_{\mathbf{i}};\lambda) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 - \lambda}{\epsilon_0}\right), \tag{13}$$

$$\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2 \approx \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \sqrt{\|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2 + \epsilon_1}. \tag{14}$$

The approximations become exact if the parameters ϵ_0 in (13) and ϵ_1 in (14) go to 0^+.

Though random walk metropolis does not rely on the assumption of smooth posterior, its low efficiency makes it impractical to apply in high-dimensional problems. We compare MALA and the random walk metropolis in Section A.7.3 of the Appendix through a simulation experiment. The experiment demonstrates the advantage of MALA, and it is worthwhile to smooth the likelihood and prior. The idea of approximating the non-differentiable thresholding function and ℓ_1-norm by smooth ones is commonly used in many areas such as spiking

neural networks (Bohte et al., 2000) and brain-machine interface technology (Onaran et al., 2013). Another advantage of using the smooth approximation is to improve the computational efficiency of MCMC (see, e.g., Rischard et al., 2018). Furthermore, our approximations (13) and (14) can be interpreted as Student t smoothing with 1 and 2 degrees of freedom, respectively. This is similar to the Gaussian smoothing technique of Chatterji et al. (2020). Details of these smoothing representations are provided in Section A.4 of the Appendix.

Algorithm A.1 in Section A.5.1 of the Appendix presents the details of the posterior updates. With the training sample size N, the computational complexity of our algorithm is $O(NpK+pK^2)$, where p is the number of entries of the tensor covariate (i.e., $p = P_1P_2\cdots P_D$ for a D-way tensor) and K is the dimension of the spline bases. After the algorithm execution, the active regions are determined by the estimated receiver operating characteristic (ROC) curve (Hajian-Tilaki, 2013) according to the posterior samples of BFEN, which is also provided in Section A.5.1 of the Appendix. The posterior point estimator $\widehat{f}_{\mathbf{i}}$ of the additive component function in the active regions is computed by $\boldsymbol{\phi}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{\mathbf{i}}$ where $\widehat{\boldsymbol{\beta}}_{\mathbf{i}}$ is the posterior mean of the truncated spline coefficients (7). Overall, our method includes several hyperparameters $\{r, \rho_{\boldsymbol{\alpha}}, p_0, p_1, \sigma_\mu^2, p, a, b\}$ and tuning parameters $\{\delta', \epsilon_0, \epsilon_1\}$ in (10)–(14). For ease of tuning, we suggest to standardize the responses in practice. After this, we assign $(2P_1\cdots P_D - 1)K/2$ to p_1 as discussed in Section 3.4 and a small number 10^{-6} to ϵ_1. The choice of ϵ_0 is data-driven and addressed in Section A.5.2 of the Appendix. We find that our model is not sensitive to the specific choice of small value for ϵ_1 through a sensitivity analysis in Section A.5.2 of the Appendix. We also suggest to set $\delta' = 0.0001$ in prior (11), and set $\sigma_\mu^2 = 100$, $p = 1$ and $a = b = 0.5$ in prior (10). The sensitivity analyses of these parameters are presented in Section A.5.4 of the Appendix. As for $(r, \rho_{\boldsymbol{\alpha}})$ in the prior (11) and p_0 in the hyperprior (12), a validation method is suggested since they are critical in controlling the strength of the prior. In our experiments, we split the available data into a training set and a validation set with sizes in the ratio of 5 to 1, and the optimal parameters are those minimizing the validation loss $\mathrm{L}(\mathbf{y}_{\text{valid}}, \widehat{\mathbf{y}}_{\text{valid}}) := (1/N_{\text{valid}})\|\mathbf{y}_{\text{valid}} - \widehat{\mathbf{y}}_{\text{valid}}\|_2^2$, where N_{valid} is the size of the validation set, and $\widehat{\mathbf{y}}_{\text{valid}}$ is the vector of predicted values of the observations $\mathbf{y}_{\text{valid}}$ in the validation set. The details of this procedure are discussed in Section A.5.3 of the Appendix. We find that the above strategy of selecting the hyperparameters works

reasonably well in all of our numerical experiments.

4 Simulation

In this section we compare our method, Bayesian additive tensor regression with FEN prior (BFEN), with three alternative methods: i) the sparse nonparametric tensor additive regression (STAR) with the group lasso penalty (Hao et al., 2021); ii) the frequentist linear tensor regression (FTR) with the lasso penalty (Zhou et al., 2013); iii) the Bayesian linear tensor regression (BTR, Guhaniyogi et al., 2017).

4.1 Simulation Settings

In our simulation study, the covariate $\mathbf{X}$ is a 2-way tensor (i.e., matrix) of dimension $P_1 \times P_2$, and so the corresponding additive model can be written as $f(\mathbf{X}) = \mu + \sum_{i,j} f_{ij}(X_{ij})$ where many f_{ij}'s are identically zero.

We let $\mu = 0$ and consider three different patterns of true active regions (non-zero additive component functions): low-rank shapes, a horse shape, and a shape of handwritten Arabic six from MNIST database (LeCun, 1998). These patterns are depicted in Figure 5 where the non-black pixels indicate the positions of the active regions.

Each pattern includes two nonlinear settings with different levels of signal-to-noise ratio SNR $= 5$ and SNR $= 50$ respectively, and one linear setting with SNR $= 5$. In the nonlinear settings, for each pixel (i, j) in the true active regions, we set $f_{ij}(x) = h_{ij}(x) - m_{ij}$ with

$$h_{ij}(x) = a_{ij} \sin(c_{ij}x) + a_{ij} \cos(d_{ij}x) + b_{ij}x, \tag{15}$$

and $m_{ij} = \int h_{ij}(x)\,\mathrm{d}x$ such that $f_{ij}(x)$ is centered.

We now specify the additive component functions in the active regions through the coefficients a_{ij}, b_{ij}, c_{ij}, and d_{ij} in (15) for each setting. First, for the linear cases, we let $a_{ij} = 0$ and $b_{ij} = 1$ in all three patterns. For the nonlinear cases, we set a_{ij} for three patterns in different ways. In particular, a_{ij} is set to 1 for every pixel (i, j). For the shape of handwritten Arabic six, we let $\mathbf{W}$ be the gray-scale matrix of this figure in the MNIST database, and a_{ij} is set as $2W_{ij} + 1$. For the horse shape, we follow Dong et al. (2016) which applies the eigenvectors

Table 2: Specification of the component function coefficients a_{ij}, b_{ij}, c_{ij}, and d_{ij} in (15) for (i, j) in the true active regions for each simulation setting. The nine settings are organized into three groups by their patterns (shapes) of the active regions.

Setting ID	1	2	3	4	5	6	7	8	9
Shape	Low rank			Horse			Handwritten Arabic six		
SNR	5	50	5	5	50	5	5	50	5
Setting Meaning	low SNR nonlinear	high SNR nonlinear	linear	low SNR nonlinear	high SNR nonlinear	linear	low SNR nonlinear	high SNR nonlinear	linear
True f_{ij}	$a_{ij}\sin(c_{ij}x) + a_{ij}\cos(d_{ij}x) + b_{ij}x - m_{ij}$								
m_{ij}	$\int a_{ij}\sin(c_{ij}x) + a_{ij}\cos(d_{ij}x) + b_{ij}x\,\mathrm{d}x$								
a_{ij}	1		0	$1\overline{u}_{ij}^{(1)} + 2$		0	$2W_{ij} + 1$		0
c_{ij}	1.5π		0	$v_{ij}^{(2)}$		0	$v_{ij}^{(2)}$		0
d_{ij}	1.5π		0	$v_{ij}^{(3)}$		0	$v_{ij}^{(3)}$		0
b_{ij}	$\frac{2}{\pi}a_{ij}(c_{ij} + d_{ij})$		1	$\frac{2}{\pi}a_{ij}(c_{ij} + d_{ij})$		1	$\frac{2}{\pi}a_{ij}(c_{ij} + d_{ij})$		1

of the graph Laplacian matrix to produce smooth signals on the graph. More specifically, we construct the spatially smooth coefficients a_{ij}'s based on the eigenvectors of the graph Laplacian matrix of the graph $\mathcal{G}$ defined in Section 2. As for c_{ij} and d_{ij}, we set them to 1.5π in the nonlinear cases of the low-rank shapes. For the other two shapes, c_{ij} and d_{ij} are also spatially smooth with value restricted to $[\pi, 1.5\pi]$. Then, b_{ij} is set as $(2/\pi)a_{ij}(c_{ij} + d_{ij})$ for all the nonlinear settings. Finally, we generate the noise terms by adjusting the variance to achieve SNR $= 50$ for Settings 2, 5, 8, and SNR $= 5$ for the others. Overall, we have nine simulation settings with different shapes of active regions, signal-to-noise ratios, and complexities of the nonlinear functions. These nine settings are summarized in Table 2 where the details of constructing $\overline{u}_{ij}^{(1)}$, $v_{ij}^{(2)}$, and $v_{ij}^{(3)}$ by following Dong et al. (2016) are provided in Section A.7.1 of the Appendix.

For each setting, the entries of each covariate $\mathbf{X}$ are generated from i.i.d. unif$(0, 1)$, and the response is generated from the additive model with corresponding observational noise level σ_ϵ^2. We generated 30 simulated datasets of sample size 600 independently for each setting. We apply the proposed BFEN and the alternatives on the datasets. For BFEN, the hyperparameters are selected as discussed in Section 3.5. For STAR, FTR and BTR, we implement these three methods respectively following Hao et al. (2021), Zhou et al. (2013) and Guhaniyogi et al. (2017), and the details are provided in Section A.6 of the Appendix.

To evaluate the estimation accuracy of the component functions for various methods, we calculate the mean squared error (MSE) and relative mean squared error (RMSE) as

$$\text{MSE} = \frac{1}{P_1 P_2} \sum_{i,j} \|f_{ij} - \widehat{f}_{ij}\|^2_{\mathbb{L}_2} \quad \text{and} \quad \text{RMSE} = \frac{1}{|\mathcal{V}|} \sum_{(i,j)\in\mathcal{V}} \|f_{ij} - \widehat{f}_{ij}\|^2_{\mathbb{L}_2} / \|f_{ij}\|^2_{\mathbb{L}_2},$$

where $\mathcal{V}$ is the set of indices of true active functions. The ability to select the active functions/pixels is assessed by the true positive rate (TPR) and the true negative rate (TNR). Note that the posterior samples of the BTR method do not directly indicate the activity of pixels directly. To evaluate the region selection performance of BTR, we follow Guhaniyogi et al. (2017) to identify the active pixels of BTR by checking whether the 95% posterior credible intervals exclude 0. For our proposed BFEN method, we used the posterior sample as introduced in Section 3.5. We further use the testing relative prediction error (RPE) to evaluate the prediction accuracy. To do this, we generate another 400 observations as a testing dataset whose index set is denoted by $\mathcal{T}$, and calculate

$$\text{RPE} = \sum_{n\in\mathcal{T}} (\widehat{y}_n - y_n)^2 / \sum_{n\in\mathcal{T}} y_n^2, \tag{16}$$

where $\widehat{y}_n$ the predicted value of the n-th observation in the test set through $\widehat{\mu}$ and $\widehat{f}_{ij}$'s.

4.2 Results

The results are presented visually as boxplots in Figure 4, which summarizes RPE, MSE, RMSE, TPR, and TNR based on 30 replicates for each setting. We also provide detailed numerical results of the simulation experiments in Table A.3 of the Appendix. To compare the computational efficiency between our algorithm and the alternatives, all methods were run on the same platform with a 2.2-GHz Intel E5-2650 v4 CPU and the execution time is also recorded in Table A.3. The convergence time of Algorithm A.1 for our proposed BFEN method is less than 1.5 minutes on average for a single specification of tuning parameters.

It can be seen that the average RPE, MSE, and RMSE of BFEN are smaller than those of STAR, FTR, and BTR in all settings of irregular sparsity shapes, i.e., a horse and a handwritten Arabic six (Settings 4–9). In Settings 1–3, the true active region is of low rank which is indeed in favor of the other alternative methods. It is expected that STAR

works well in Setting 2 and the linear alternatives have better performance in Setting 3 since the corresponding settings favor these models. Besides, among the three alternative methods, STAR enjoys an advantage over FTR and BTR only when nonlinear signals are strong enough (Settings 2, 5, and 8). Overall, BFEN is more flexible and has advantages in a wider range of scenarios.

As for the recovery of active regions, the proposed BFEN method has a balanced performance in both TPR and TNR for all settings. We find that STAR and FTR tend to over-select active pixels, i.e., TNR is low. On the other hand, TPR of BTR deteriorates considerably when its low rank and linear assumption are violated in Settings 4, 5, 7, and 8.

For further demonstration, we calculate the $\mathbb{L}_2$-norm of each estimated additive component function, $\|\widehat{f}_{ij}\|_{\mathbb{L}_2}$, for all methods. These results can be visualized by heatmaps for each simulated dataset. For the nonlinear with high SNR settings, the heatmap corresponding to the median RPE among 30 simulated datasets for each method was depicted in Figure 5, and the heatmap for the truth was also depicted at the leftmost of Figure 5. For the nonlinear with low SNR and linear settings, the heatmaps were respectively provided in Figures A.3 and A.4 of the Appendix. It is evident that the proposed BFEN recovers the shape of the true active region and the corresponding spatial distribution of the signal strength with a reasonably good accuracy.

In contrast, STAR, FTR, and BTR only work well in the low-rank setting (Setting 2, the first row in Figure 5) but are substantially worse for the other two patterns. STAR, FTR, and BTR are based on the tensor rank-R CP decomposition, which is the sum of R rank-1 tensors. Therefore, the sparsity patterns recovered by these methods tend to be a combination of several rectangular blocks. BFEN, however, encourages the similarity of neighbouring signals rather than enforcing certain shapes of spatially connected regions and, thus, can adaptively identify the active regions with complex shapes.

5 Facial Feature Analysis

We apply our method to the Labeled Faces in the Wild dataset (Huang et al., 2008). This dataset consists of facial images collected from $5,721$ people and attributes that quantify

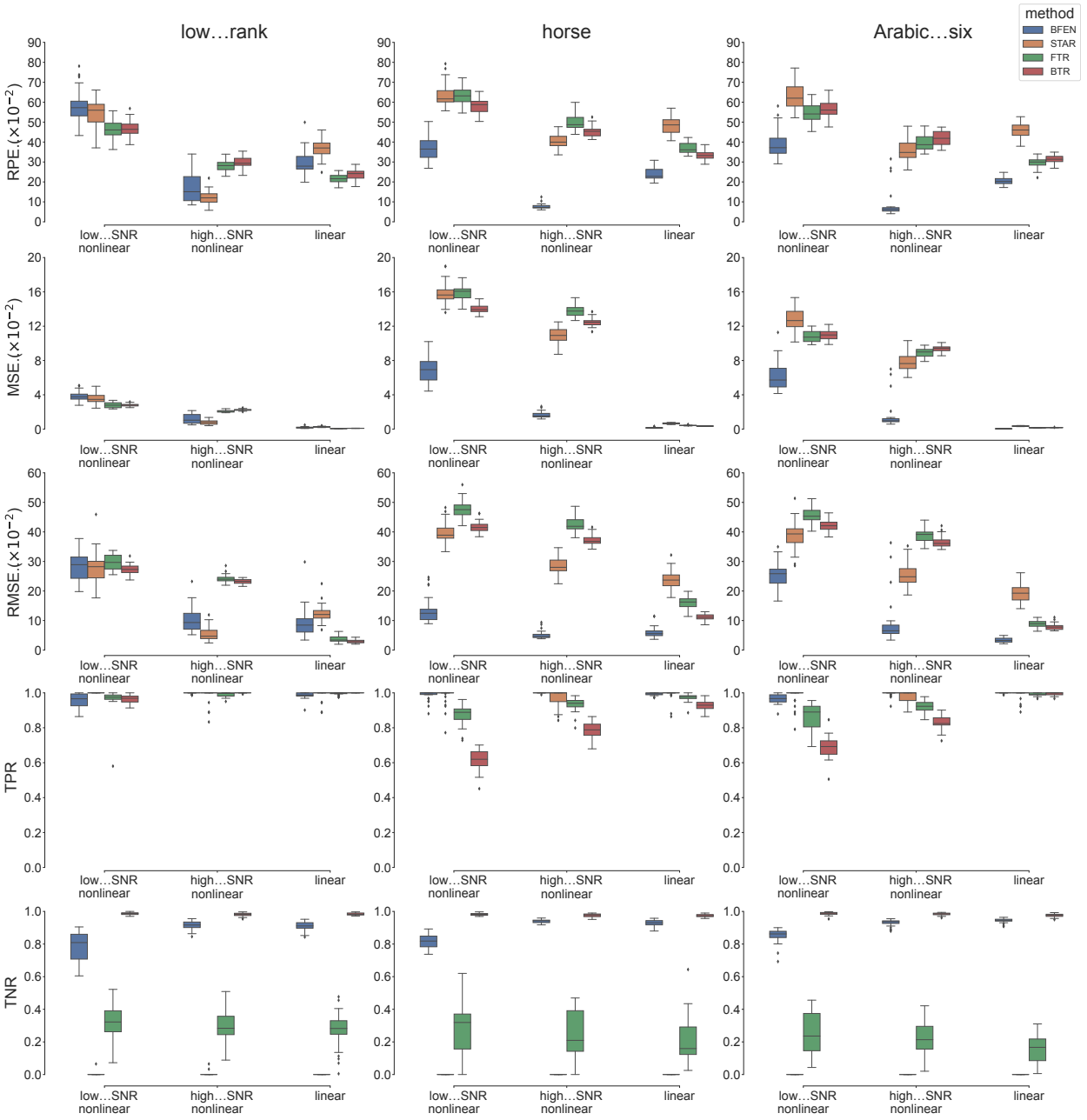

Figure 4: The boxplots for visualizing the results of simulation experiments. Rows 1–5 depict RPE, MSE, RMSE, TPR, and TNR, respectively. Columns 1–3 respectively correspond to low-rank shapes, a horse shape, and a shape of handwritten Arabic six. In each panel, the left, middle, and right group of boxes correspondingly represent the results under 'low SNR, nonlinear', 'high SNR, nonlinear' and 'linear' setting. In each setting, the blue, orange, green, and red boxes correspond to BFEN, STAR, FTR and BTR, respectively.

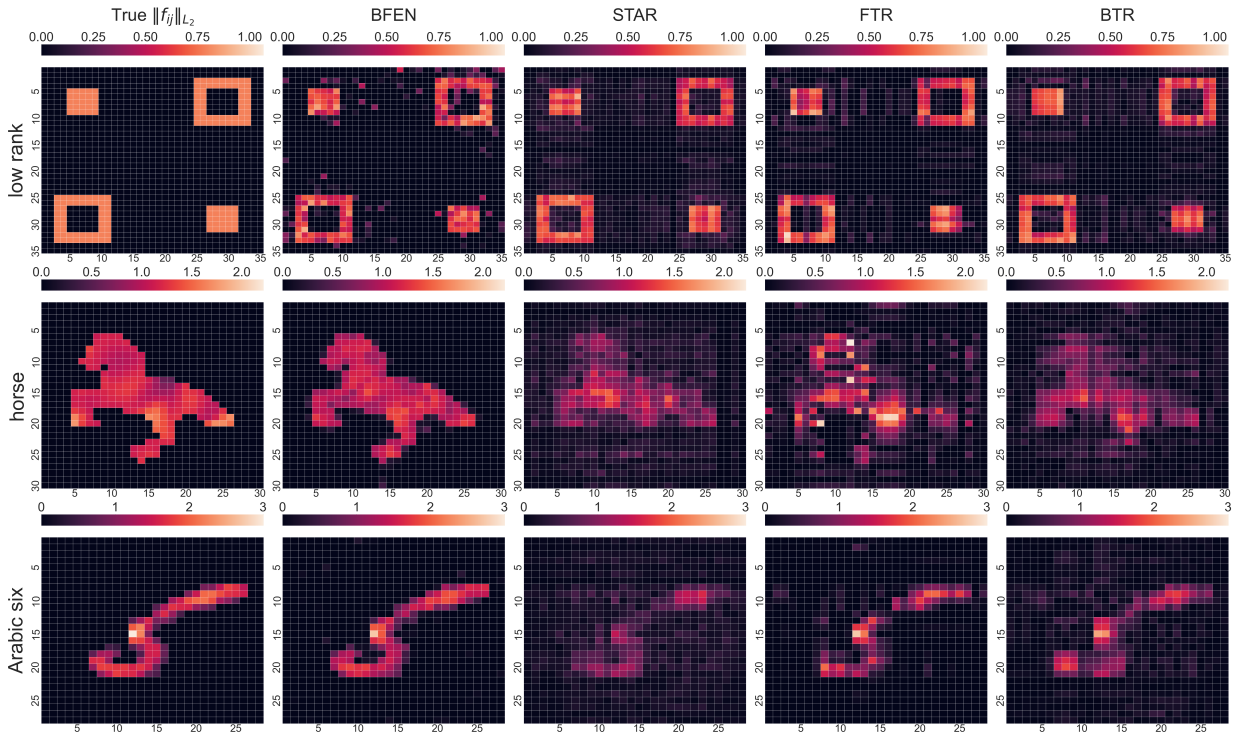

Figure 5: The heatmaps of various methods under the nonlinear with high SNR settings (Settings 2, 5, and 8). Rows 1–3 correspond to the patterns of low-rank shapes (Setting 2), a horse shape (Setting 5), and a shape of handwritten Arabic six (Setting 8), respectively. The first column presents the truth. Columns 2–5 correspond to the estimated results by BFEN, STAR, FTR, and BTR, respectively.

various facial features for each facial image (Kumar et al., 2009). In this experiment, we select one facial image per person and choose the facial expressions related to the mouth as responses, which are *smiling*, *frowning*, *mouth closed*, *mouth wide open*, and *teeth not visible*.

We follow Hassner et al. (2015) to register these images. In particular, all images are frontalized to make faces in constrained and forward-facing poses; thus, the same regions of different images represent the same part of a human face. The original gray-scale image is of size 90×90 with entry values in $[0, 255]$. We further down-sample each image to a 45×45 matrix by replacing every four pixels in a square with one pixel of average gray-scale value, and rescale the entry values to $[0, 1]$. Figure 7(a) shows an example of the resulting image.

We compare the proposed method with STAR, FTR, and BTR as in Section 4. We randomly sample an index set $\mathcal{S}$ of size 2000 from the full subject set $\{1, \cdots, 5721\}$ for

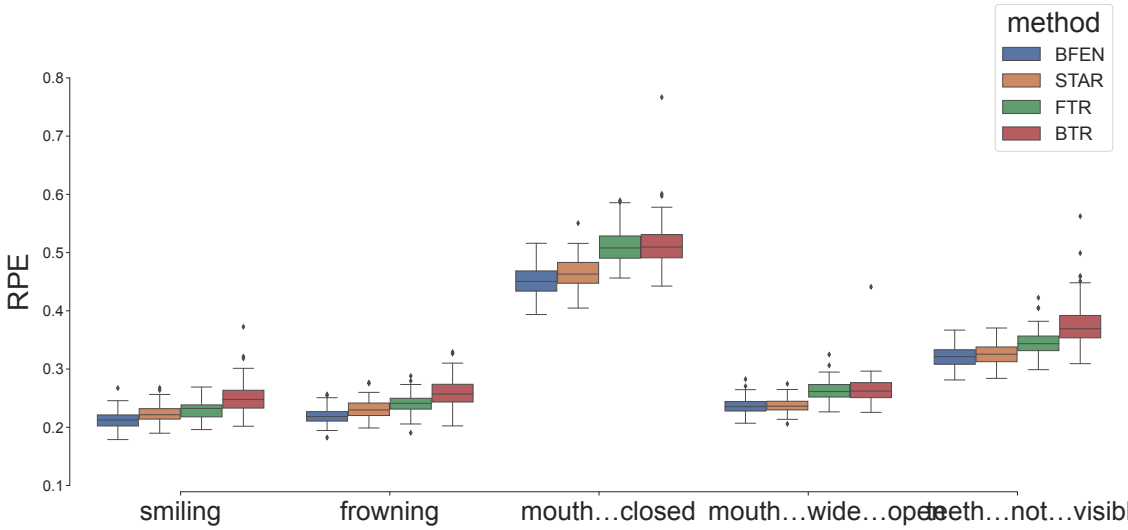

Figure 6: The boxplots of the relative predictive errors for the facial data analysis under 100 replicates. From left to right, the 5 groups of the boxes respectively represent the results for attributes *smiling*, *frowning*, *mouth closed*, *mouth wide open*, and *teeth not visible*, respectively. In each group, the blue, orange, green, and red boxes respectively correspond to BFEN, STAR, FTR, and BTR.

feasible computation. The set $\mathcal{S}$ is then divided into three disjoint subsets $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{T}$ of sizes 1000, 200 and 800 respectively. Sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are used for training and tuning, and Set $\mathcal{T}$ is used to evaluate the performance of prediction through RPE in (16). We repeat this procedure 100 times.

The results are presented visually as boxplots in Figure 6, which summarizes the RPE of various methods for each attribute. We also provide the numerical results and runtime for the facial feature analysis in Table A.5 of the Appendix. In particular, Algorithm A.1 of our proposed BFEN method needs less than 2 minutes on average to converge for one grid of tuning parameters with a 2.2-GHz Intel E5-2650 v4 CPU. It shows that BFEN outperforms the three competitors in all cases, except for the response *mouth wide open* where BFEN and STAR have similarly good performances. The heatmaps in Figure 7(c) display the magnitude $\|\widehat{f}_{ij}\|_{L_2}$ of each pixel for the attribute *smiling* using various methods. It shows that the result of BFEN has better interpretability: *smiling* can be characterized by the pixel values around the eyes, mouth and some facial muscles. Figure 7(b) depicts

the estimated nonlinear functions f_{ij}'s and the 95% posterior credible intervals by BFEN corresponding to the region indicated by the rectangle in Figure 7(a). Some functions exhibit clear non-linearity. In contrast, the signals selected by FTR and BTR do not have an obvious interpretation. With the help of nonlinearity and the group regularization across different blocks, STAR has better interpretability than that of FTR and BTR, but is still inferior to BFEN. Overall, the low-rank modeling may not be flexible enough to characterize a complex shape like smiling, and this result is consistent with our findings in the simulation study. The heatmaps for other attributes are depicted in Figure A.6 of the Appendix.

6 Discussion

In this paper, we have proposed a nonlinear Bayesian tensor additive regression model, which incorporates the spatial information of the tensor covariates. A functional version of the fused elastic net, FEN, has been introduced as a prior distribution on the additive component functions to accommodate the sparse, spatially smooth functional structure with discontinuous jumps. Through numerical experiments on the simulated and the facial feature datasets, we have demonstrated the superior performance of the proposed method compared to the existing linear and nonlinear tensor regression models for characterizing irregular shapes of sparse active regions, even if the signal-to-noise ratio is relatively low. The performance of alternative methods, however, rely on low-rank assumption, which is often violated in real applications of image and neuroscience data.

The proposed BFEN has some limitations, which may lead to extension of this work. Similar to many other methods with multiple hyperparameters, the main computational burden of our method is due to the validation method for selecting hyperparameters p_0, r, and $\rho_{\boldsymbol{\alpha}}$. Further investigation is needed to relieve this bottleneck by, for example, imposing appropriate hyperpriors on these hyperparamters to automatically adjust them. In addition, extending the current model to the case of multi-dimensional response variables, like matrix-on-tensor and tensor-on-tensor regressions, is also of interest.

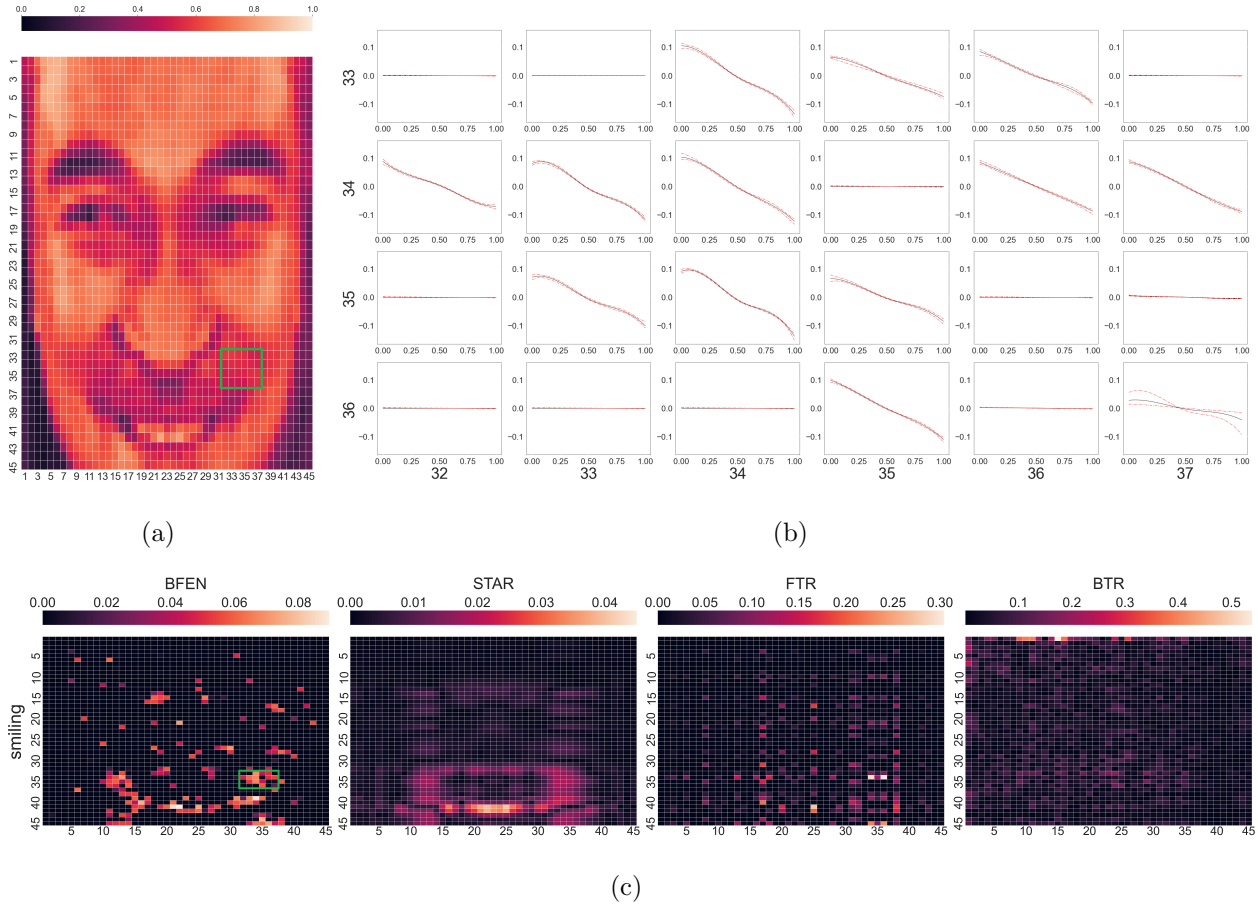

Figure 7: Real applications on the facial data for the attribute *smiling*. (a) An example of facial covariate tensor $\mathbf{X}$. Its corresponding value of smiling attribute is 1.51, which means the person is smiling. (b) Each of the 4×6 panels depicts the estimated $\widehat{f}_{ij}$ from BFEN corresponding to the enclosed rectangle area in (a), and between the dashed lines are the 95% posterior credible intervals. (c) The heatmaps in columns 1–4 correspond to $\|\widehat{f}_{ij}\|_{\mathbb{L}_2}$ estimated by BFEN, STAR, FTR, and BTR, respectively.

Appendix

A.1 Construction of Spline Basis

In this section, we show the details of the basis construction in Section 3.3. The K dimensional centered and orthonormal basis $\boldsymbol{\phi}$ is from an $K+1$ dimensional B-spline basis $\boldsymbol{\psi}$. The construction is divided into three steps:

First, denote $\mathbf{W} := \int \boldsymbol{\psi}(x)\boldsymbol{\psi}(x)^{\mathrm{T}}\,\mathrm{d}x$. Suppose its eigendecomposition is $\mathbf{W} = \mathbf{V}\boldsymbol{\Gamma}_1\mathbf{V}^{\mathrm{T}}$, where $\boldsymbol{\Gamma}_1$ is diagonal containing the eigenvalues and $\mathbf{V}$ is orthonormal containing the eigenvectors in its columns. Set $\tilde{\boldsymbol{\psi}}(x) := \boldsymbol{\Gamma}_1^{-1/2}\mathbf{V}^{\mathrm{T}}\boldsymbol{\psi}(x)$ to get an orthonormal basis satisfying

$$\int \tilde{\boldsymbol{\psi}}(x)\tilde{\boldsymbol{\psi}}(x)^{\mathrm{T}}\,\mathrm{d}x = \mathbf{I}_{K+1}.$$

Next, denote $\mathbf{d} := \int \tilde{\boldsymbol{\psi}}(x)\,\mathrm{d}x \in \mathbb{R}^{K+1}$, set $\mathbf{T} \in \mathbb{R}^{K+1,K}$ as the full column rank matrix with columns orthornormal to $\mathbf{d}$, i.e., $\mathbf{T}^{\mathrm{T}}\mathbf{d} = \mathbf{0}$. Set $\tilde{\boldsymbol{\phi}}(\cdot) := \mathbf{T}^{\mathrm{T}}\tilde{\boldsymbol{\psi}}(x)$, and we get an set of orthonormal and centered basis functions satisfying

$$\int \tilde{\boldsymbol{\phi}}(\cdot)\tilde{\boldsymbol{\phi}}(\cdot)^{\mathrm{T}}\,\mathrm{d}x = \mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathbf{I}_K \quad \text{and} \quad \int \tilde{\boldsymbol{\phi}}(x)\,\mathrm{d}x = \mathbf{0}.$$

Finally, denote $\boldsymbol{\Omega}_0 := \int \tilde{\boldsymbol{\phi}}''(x)\tilde{\boldsymbol{\phi}}''(x)^{\mathrm{T}}\,\mathrm{d}x$. Suppose it has the eigendecomposition $\boldsymbol{\Omega}_0 = \mathbf{U}\boldsymbol{\Gamma}_2\mathbf{U}^{\mathrm{T}}$, where $\boldsymbol{\Gamma}_2$ is a diagonal matrix of eigenvalues arranged in an increasing order. Set $\boldsymbol{\phi}(x) := \mathbf{U}^{\mathrm{T}}\tilde{\boldsymbol{\phi}}(x)$. We can see that $\boldsymbol{\phi}$ is a centered and orthornormal basis with a diagonal

$$\boldsymbol{\Omega} := \int \boldsymbol{\phi}''(x)\boldsymbol{\phi}''(x)^{\mathrm{T}}\,\mathrm{d}x = \mathbf{U}^{\mathrm{T}}\boldsymbol{\Omega}_0\mathbf{U} = \mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\Gamma}_2\mathbf{U}^{\mathrm{T}}\mathbf{U} = \boldsymbol{\Gamma}_2.$$

Hence the properties (i), (ii), and (iii) in Section 3.3 of the main paper are all satisfied by the constructed spline basis $\boldsymbol{\phi}$.

As $\boldsymbol{\Omega}_0$ is computed from the second order derivative of $\tilde{\boldsymbol{\phi}}$, its smallest eigenvalue is zero, i.e., $(\boldsymbol{\Gamma}_2)_{11} = 0$. This eigenvalue corresponds to an eigenvector $\mathbf{u}_1$ such that $\mathbf{u}_1^{\mathrm{T}}\tilde{\boldsymbol{\phi}}$ is a linear function. This implies the first component of $\boldsymbol{\phi}$ is a linear function. Because the properties (i) and (ii) in Section 3.3 are both satisfied, it is easy to see that $\phi_k \perp \ell[0,1], k > 1$. Meanwhile, it can be verified that $\phi_1\boldsymbol{\alpha}_{\mathbf{i}1}$ is the projection of the function $g_{\mathbf{i}} = \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}}$ onto $\ell[0,1]$, i.e., $\mathcal{P}g_{\mathbf{i}} = \boldsymbol{\phi}_1\boldsymbol{\alpha}_{\mathbf{i}1}$, where $\boldsymbol{\alpha}_{\mathbf{i}1}$ is the first element of $\boldsymbol{\alpha}_{\mathbf{i}}$. Hence, with the basis $\boldsymbol{\phi}$, the roughness norm $\mathcal{R}(g_{\mathbf{i}}) = \|g_{\mathbf{i}}''\|_{\mathbb{L}_2}^2 + \delta'\|\mathcal{P}g_{\mathbf{i}}\|_{\mathbb{L}_2}^2$ in (4) of the main paper can be rewritten as $\sum_{\mathbf{i}\in\mathcal{I}}\boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\mathbf{R}\boldsymbol{\alpha}_{\mathbf{i}}$ where $\mathbf{R} = \boldsymbol{\Omega} + \delta'\|\phi_1\|_{\mathbb{L}_2}^2\mathbf{e}_1\mathbf{e}_1^{\mathrm{T}}$ with $\mathbf{e}_1 = (1,0,\ldots,0)^{\mathrm{T}}$.

Using the above procedure, constructing the orthonormal basis only requires to know the degree and dimension of the B-spline. For the degree, it can be fixed as 4 (cubic spline) to alleviate the computational burden, and this choice is commonly used in nonparametric literature Huang et al. (2010). As for K, there are many simple recommendations based on the sample size (e.g., Ruppert et al., 2003). We follow Fan et al. (2011) to fix $K = \lceil n^{1/5} \rfloor$, where $\lceil\cdot\rfloor$ denotes rounding to the nearest integer, n is the sample size, and the interior knots are equally-spaced quantiles of all covariate samples. All the results of numerical experiments exhibited in our paper are obtained through this empirical rule, and we find that this rule works reasonably well.

A.2 The Order of Normalizing Term of the Prior

In this section, we provide the proof of Proposition 1 by calculating the orders of the normalizing term C_{δ,σ^2} of the prior (11) with respect to δ and σ^2 respectively.

The normalizing term C_{δ,σ^2} equals to

$$\int_{\mathbb{R}^{P_1\cdots P_D K}} \exp\left\{ -\frac{\delta \sum_{\mathbf{i}\in\mathcal{I}} \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}} \mathbf{R} \boldsymbol{\alpha}_{\mathbf{i}}}{\sigma^2} - \frac{r\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2 + (1-r)\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2}{2\sigma^2\rho_{\boldsymbol{\alpha}}} \right\} \mathrm{d}\boldsymbol{\alpha} \tag{A.1}$$

With a formula

$$\exp\left(-\frac{\lambda|a|}{\sigma}\right) = \int_0^\infty \frac{\lambda}{\sqrt{2\pi\omega^2}} \exp\left(-\frac{a^2}{2\sigma^2\omega^2} - \frac{\lambda^2\omega^2}{2}\right) \mathrm{d}\omega^2$$

for $(\lambda > 0)$ (Andrews and Mallows, 1974), we have

$$\exp\left\{ -\frac{(1-r)\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2}{2\sigma^2\rho_{\boldsymbol{\alpha}}} \right\} = \prod_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \int_0^\infty \frac{1-r}{\sqrt{2\pi\omega_{\mathbf{i}\mathbf{i}'}^2}} \cdot \exp\left\{ -\frac{\|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2}{2\cdot 4\sigma^4\rho_{\boldsymbol{\alpha}}^2\omega_{\mathbf{i}\mathbf{i}'}^2} - \frac{(1-r)^2\omega_{\mathbf{i}\mathbf{i}'}^2}{2} \right\} \mathrm{d}\omega_{\mathbf{i}\mathbf{i}'}^2 . \tag{A.2}$$

Here we introduce some notations to facilitate the proof. Let $\boldsymbol{\alpha}_{\cdot k}$ denote the tensor of dimension $P_1 \times \cdots \times P_D$ whose $\mathbf{i}$-th element, $(\boldsymbol{\alpha}_{\cdot k})_{\mathbf{i}}$, is $\boldsymbol{\alpha}_{\mathbf{i}k}$ (the k-th element of coefficient vector $\boldsymbol{\alpha}_{\mathbf{i}}$), $\mathbf{i} = (i_1, \ldots, i_D) \in \mathcal{I}$ and $k = 1, \ldots, K$. For a generic D-way tensor, we define the operator $\mathrm{vec}(\cdot)$ as its vectorization according to the lexicographical order from its 1-st to D-th mode. In other words, $\mathrm{vec}(\mathcal{I}) := (1, 2, \cdots, \prod_{d=1}^D P_d)$ as the vectorized form of the index set $\mathcal{I}$ such that $\mathbf{i} = (i_1, \ldots, i_D)$ is now placed at the t-th element of $\mathrm{vec}(\mathcal{I})$, where $t = i_1 + \sum_{d=2}^D (i_d - 1)\prod_{d'=1}^{d-1} P_{d'}$, for any $\mathbf{i} \in \mathcal{I}$. Similarly, $\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k})$ is the vectorized $\boldsymbol{\alpha}_{\cdot k}$ such that the t-th element of $\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k})$ is $(\boldsymbol{\alpha}_{\cdot k})_{\mathbf{i}}$ whenever $t = i_1 + \sum_{d=2}^D (i_d - 1)\prod_{d'=1}^{d-1} P_{d'}$. With the vectoried $\boldsymbol{\alpha}_{\cdot k}$, some terms in the integrand of (A.1) can be rewritten in terms of quadratic forms. In particular, define the quadratic matrices $\boldsymbol{\Lambda}_1^{(k)}$ for $k = 1, \ldots, K$, $\boldsymbol{\Lambda}_2$, and $\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}$ as follows. $\boldsymbol{\Lambda}_1^{(k)}$ is $G_{kk} \cdot \mathbf{I}$, where $G_{kk} \in \mathbb{R}$ is the k-th diagonal element of the matrix $\mathbf{R}$ in (A.1). For any edge $(\mathbf{i}, \mathbf{i}') \in \mathcal{E}$ of the graph, suppose $\mathbf{i}$ (or $\mathbf{i}'$) corresponds to the t-th (or t'-th resp.) element of $\mathrm{vec}(\mathcal{I})$, the (t,t')-th and (t,t)-th elements of $\boldsymbol{\Lambda}_2$ and $\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}$ are $(\boldsymbol{\Lambda}_2)_{tt'} = -1/(2\rho_{\boldsymbol{\alpha}})$, $(\boldsymbol{\Lambda}_2)_{tt} = -\sum_{s\neq t}(\boldsymbol{\Lambda}_2)_{ts}$, $(\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})})_{tt'} = -1/(2\cdot 4\rho_{\boldsymbol{\alpha}}^2\omega_{\mathbf{i}\mathbf{i}'}^2)$, and $(\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})})_{tt} = -\sum_{s\neq t}(\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})})_{ts}$; The other elements of $\boldsymbol{\Lambda}_2$ and $\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}$ are 0's. The three matrices satisfy:

$$\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k})^{\mathrm{T}} \boldsymbol{\Lambda}_1^{(k)} \mathrm{vec}(\boldsymbol{\alpha}_{\cdot k}) = \delta \sum_{\mathbf{i}\in\mathcal{I}} G_{kk}\alpha_{\mathbf{i}k}^2, \tag{A.3}$$

$$\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k})^{\mathrm{T}} \boldsymbol{\Lambda}_2 \mathrm{vec}(\boldsymbol{\alpha}_{\cdot k}) = \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \frac{(\alpha_{\mathbf{i}k} - \alpha_{\mathbf{i}'k})^2}{2\rho_{\boldsymbol{\alpha}}}, \tag{A.4}$$

$$\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k})^{\mathrm{T}}\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k}) = \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \frac{(\alpha_{\mathbf{i}k}-\alpha_{\mathbf{i}'k})^2}{2\cdot 4\rho_{\boldsymbol{\alpha}}^2\omega_{\mathbf{i}\mathbf{i}'}^2}. \tag{A.5}$$

Denote

$$g(\boldsymbol{\omega}) = \prod_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \frac{1-r}{\sqrt{2\pi\omega_{\mathbf{i}\mathbf{i}'}^2}} \exp\left\{-\frac{(1-r)^2\omega_{\mathbf{i}\mathbf{i}'}^2}{2}\right\}$$

and

$$\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2) = \boldsymbol{\Lambda}_1^{(k)} + \frac{r\boldsymbol{\Lambda}_2}{\delta} + \frac{\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}}{\delta\sigma^2}.$$

After using the above simplified notations, we substitute (A.2) back into (A.1), and apply the Fubini's Theorem to have

$$C_{\delta,\sigma^2} = \int_{R_+^{|\mathcal{E}|}} g(\boldsymbol{\omega}) \left[\prod_{k=1}^{K} \int_{\mathbb{R}^{P_1\cdots P_D}} \exp\left\{-\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k})^{\mathrm{T}} \frac{\delta\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)}{\sigma^2}\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k})\right\} \mathrm{d}\boldsymbol{\alpha}_{\cdot k}\right] \mathrm{d}\boldsymbol{\omega}^2.$$

Integrating out $\boldsymbol{\alpha}_{\cdot k}$'s, the normalizing term C_{δ,σ^2} becomes

$$c\int_{R_+^{|\mathcal{E}|}} g(\boldsymbol{\omega}) \left(\frac{\sigma^2}{\delta}\right)^{P_1\cdots P_D K/2} \left[\prod_{k=1}^{K} \sqrt{\det\{\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)\}}\right]^{-1} \mathrm{d}\boldsymbol{\omega}^2. \tag{A.6}$$

Hence, the key to compute the degrees of δ and σ^2 is to find out the degrees within $\det\{\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)\}$. Since $\mathcal{G} = (\mathcal{I},\mathcal{E})$ is a connected graph, (A.4) and (A.5) equal to 0 if and only if $\boldsymbol{\alpha}_{\mathbf{i}k} = \boldsymbol{\alpha}_{\mathbf{i}'k}, \forall \mathbf{i},\mathbf{i}' \in \mathcal{I}$, i.e. $\mathrm{vec}(\boldsymbol{\alpha}_{\cdot k}) \propto \mathbf{1}$, where $\mathbf{1}$ is a $P_1\cdots P_D$ dimensional vector with each entry being 1. Hence $\boldsymbol{\Lambda}_2$ and $\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}$ are positive semidefinite matrix with rank $P_1\cdots P_D - 1$. Next we discuss the order of C_{δ,σ^2} with respect to δ,σ^2 when they go to both 0 or ∞. On one hand,

(i) $\delta\to\infty$. With the positive (semi-)definiteness of the matrices, we have

$$0 < \det\left(\boldsymbol{\Lambda}_1^{(k)}\right) \le \det\{\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)\} \le \det\left(\boldsymbol{\Lambda}_1^{(k)} + r\boldsymbol{\Lambda}_2 + \frac{\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}}{\sigma^2}\right), \quad \forall\delta > 1.$$

We can see that $\det\{\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)\}$ is bounded by two positive constants that is independent to δ, so $\det\{\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)\}$ is of order $O(1)$ for $\delta\to\infty$. Together with (A.6), we know the normalizing term C_{δ,σ^2} is of order $(1/\delta)^{P_1\cdots P_D K/2}$.

(ii) $\sigma^2\to\infty$. Similarly, we have

$$0 < \det\left(\boldsymbol{\Lambda}_1^{(k)} + \frac{r\boldsymbol{\Lambda}_2}{\delta}\right) \le \det\{\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)\} \le \det\left(\boldsymbol{\Lambda}_1^{(k)} + \frac{r\boldsymbol{\Lambda}_2}{\delta} + \frac{\boldsymbol{\Lambda}_3^{(\boldsymbol{\omega})}}{\delta}\right), \forall\sigma^2 > 1,$$

so $\det\{\boldsymbol{\Lambda}(k,\boldsymbol{\omega},\delta,\sigma^2)\}$ is of order $O(1)$ for $\sigma^2\to\infty$. Combining with (A.6), the normalizing term C_{δ,σ^2} is of order $(\sigma^2)^{P_1\cdots P_D K/2}$.

On the other hand, to compute the degrees of δ and σ when they go to 0, we apply Grinberg (2020)'s formula for n dimensional square matrices $\mathbf{A}, \mathbf{B}$:

$$\det(\mathbf{A}+x\mathbf{B}) = \det(\mathbf{A})+\det(\mathbf{A})p_1(\mathbf{A}^{-1}\mathbf{B})x+\cdots+\det(\mathbf{A})p_{n-1}(\mathbf{A}^{-1}\mathbf{B})x^{n-1}+\det(\mathbf{B})x^n, \quad \text{(A.7)}$$

where $\mathbf{A}$ is an invertible square matrix, and $p_1(\cdot), \cdots, p_{n-1}(\cdot)$ are the sums of all principal minors of order $2, \cdots, n-1$, respectively.

(iii) $\delta \to 0$. After respectively substituting the three variates $\mathbf{\Lambda}_1^{(k)}$, $r\mathbf{\Lambda}_2 + \mathbf{\Lambda}_3^{(\boldsymbol{\omega})}/\sigma^2$, and $1/\delta$ for $\mathbf{A}$, $\mathbf{B}$, and x in formula (A.7), it shows that n in (A.7) turns to be $P_1 \cdots P_D$, and $\mathbf{A}$ is a positive definite matrix. It is easy to see that the right side of the linear combination $r \times$ (A.4) + (A.5)$/\sigma^2$ equal to 0 if and only if $\boldsymbol{\alpha}_{\mathbf{i}k} = \boldsymbol{\alpha}_{\mathbf{i}'k}$, $\forall \mathbf{i}, \mathbf{i}' \in \mathcal{I}$, so $\mathbf{B}$ is a positive semidefinite matrix with rank $P_1 \cdots P_D - 1$, and meanwhile $\mathbf{A}^{-1}\mathbf{B}$ is also positive semidefinite matrix with rank $P_1 \cdots P_D - 1$. Hence we have

$$\det\{\mathbf{\Lambda}(k, \boldsymbol{\omega}, \delta, \sigma^2)\} = \det(\mathbf{A}) + \det(\mathbf{A})p_1(\mathbf{A}^{-1}\mathbf{B})/\delta + \cdots + \det(\mathbf{A})p_{n-1}(\mathbf{A}^{-1}\mathbf{B})/\delta^{n-1},$$

where $\det(\mathbf{B}) = 0$, $\det(\mathbf{A}) > 0$, and $p_j\big(\mathbf{A}^{-1}\mathbf{B}\big) > 0$, $j = 1, \cdots, n-1$. So we know that $\det\{\mathbf{\Lambda}(k, \boldsymbol{\omega}, \delta, \sigma^2)\}$ is of order $1/\delta^{P_1 \cdots P_D - 1}$ for $\delta \to 0$. Combining with (A.6), the normalizing term C_{δ,σ^2} is of order $(1/\delta)^{K/2}$.

(iv) $\sigma^2 \to 0$. Substitute $\mathbf{\Lambda}_1^{(k)} + r\mathbf{\Lambda}_2/\delta$, $\mathbf{\Lambda}_3^{(\boldsymbol{\omega})}/\delta$, $1/\sigma^2$ for $\mathbf{A}$, $\mathbf{B}$, x in the formula (A.7), respectively. Similarly, we can prove $\det\{\mathbf{\Lambda}(k, \boldsymbol{\omega}, \delta, \sigma^2)\}$ is of order $(1/\sigma^2)^{P_1 \cdots P_D - 1}$. Further, according to (A.6), the normalizing term C_{δ,σ^2} is of order $(\sigma^2)^{(2P_1 \cdots P_D - 1)K/2}$.

A.3 Model Invariance

In this section, we prove Proposition 2 in Section 3.4 of the main paper. Denote $p(\boldsymbol{\alpha}; \boldsymbol{\phi})$ the probability distribution function of the prior (11), where '$\boldsymbol{\phi}$' is involved because $\mathbf{R}$ in the prior (11) is defined based on the spline basis. Note $f_{\mathbf{i}}(X_{\mathbf{i}}) = \boldsymbol{\phi}(X_{\mathbf{i}})^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{1}_{\{\|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 > \lambda\}}$, so the prior $p(\boldsymbol{\alpha}; \boldsymbol{\phi})$ of paramater $\boldsymbol{\alpha}$ induces a prior distribution for $f(\mathbf{X})$. For any orthogonal matrix $\mathbf{Q} \in O(K)$, after giving orthogonal transformation to the spline basis: $\boldsymbol{\phi}_{\mathbf{Q}} = \mathbf{Q}\boldsymbol{\phi}$, we have a new model $Y = f_{\mathbf{Q}}(X) + \epsilon$ with component function

$$f_{\mathbf{Q},\mathbf{i}}(X_{\mathbf{i}}) = \boldsymbol{\phi}_{\mathbf{Q}}(X_{\mathbf{i}})^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{1}_{\{\|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 > \lambda\}}. \quad \text{(A.8)}$$

With the new spline basis $\boldsymbol{\phi}_{\mathbf{Q}}$, the prior imposed on $\boldsymbol{\alpha}$ turns to be $p(\boldsymbol{\alpha}; \boldsymbol{\phi}_{\mathbf{Q}})$, which also induces a prior distribution for $f_{\mathbf{Q}}(\mathbf{X})$. Proposition 2 is an equivalent to: The prior distribution of $f_{\mathbf{Q}}(\mathbf{X})$ induced by the prior $p(\boldsymbol{\alpha}; \boldsymbol{\phi}_{\mathbf{Q}})$ keeps unchange for any orthonormal matrix

$\mathbf{Q} \in O(K)$. Denote $\boldsymbol{\alpha}_{\mathbf{Q}} \in \mathbb{R}^{P_1 \times \cdots \times P_D \times K}$ whose element $(\boldsymbol{\alpha}_{\mathbf{Q}})_{\mathbf{i}k} = (\mathbf{Q}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}})_k$, $\mathbf{i} \in \mathcal{I}, 1 \le k \le K$. Because

$$f_{\mathbf{Q},\mathbf{i}}(X_{\mathbf{i}}) = \boldsymbol{\phi}(X_{\mathbf{i}})^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{1}_{\{\|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 > \lambda\}} = \boldsymbol{\phi}(X_{\mathbf{i}})^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{1}_{\{\|\mathbf{Q}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 > \lambda\}},$$

(A.8) with the prior $p(\cdot; \boldsymbol{\phi}_{\mathbf{Q}})$ for $\boldsymbol{\alpha}$ is equivalent to

$$f_{\mathbf{Q},\mathbf{i}}(X_{\mathbf{i}}) = \boldsymbol{\phi}(X_{\mathbf{i}})^{\mathrm{T}}(\boldsymbol{\alpha}_{\mathbf{Q}})_{\mathbf{i}} \cdot \mathbf{1}_{\{\|(\boldsymbol{\alpha}_{\mathbf{Q}})_{\mathbf{i}}\|_2^2 > \lambda\}} \tag{A.9}$$

with a prior $q(\cdot; \boldsymbol{\phi}_{\mathbf{Q}})$ for $\boldsymbol{\alpha}_{\mathbf{Q}}$, where $q(\cdot; \boldsymbol{\phi}_{\mathbf{Q}})$ is obtained through density transformation from $p(\boldsymbol{\alpha}; \boldsymbol{\phi}_{\mathbf{Q}})$. All we need is to show that $q(\cdot; \boldsymbol{\phi}_{\mathbf{Q}})$ is invariant to $\mathbf{Q} \in O(K)$, which, is guaranteed by the following computation

$$\begin{aligned}
\ln q(\boldsymbol{\alpha}; \boldsymbol{\phi}_{\mathbf{Q}}) \propto & -\sum_{\mathbf{i}} \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}(\mathbf{Q}\mathbf{R}\mathbf{Q}^{\mathrm{T}})\mathbf{Q}\boldsymbol{\alpha}_{\mathbf{i}} - \frac{r\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}}\left(\boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}_{\mathbf{i}} - 2\boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}_{\mathbf{i}} + \boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}_{\mathbf{i}'}\right)}{2\sigma^2\rho_{\boldsymbol{\alpha}}} \\
& - \frac{(1-r)\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}}\sqrt{\boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}_{\mathbf{i}} - 2\boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}_{\mathbf{i}} + \boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\alpha}_{\mathbf{i}'}}}{2\sigma^2\rho_{\boldsymbol{\alpha}}} \\
\propto & -\sum_{\mathbf{i}} \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\mathbf{R}\boldsymbol{\alpha}_{\mathbf{i}} - \frac{r\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}}\left(\boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} - 2\boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} + \boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}'}\right)}{2\sigma^2\rho_{\boldsymbol{\alpha}}} \\
& - \frac{(1-r)\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}}\sqrt{\boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} - 2\boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}} + \boldsymbol{\alpha}_{\mathbf{i}'}^{\mathrm{T}}\boldsymbol{\alpha}_{\mathbf{i}'}}}{2\sigma^2\rho_{\boldsymbol{\alpha}}} \\
\propto & \, p(\boldsymbol{\alpha}; \boldsymbol{\phi}).
\end{aligned}$$

A.4 Student t Smoothing

In this section, we show that the proposed approximations (13) and (14) of nonsmooth functions in the main paper can be represented in terms of smoothing method similar to Chatterji et al. (2020), where a Gaussian smoothing was introduced.

We first recall that $\boldsymbol{\alpha}^{\mathrm{T}} = (\boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}})_{\mathbf{i}\in\mathcal{I}} \in \mathbb{R}^{P_1 \times \cdots \times P_D \times K}$ are the combined spline coefficients for all additive component functions $f_{\mathbf{i}}$, $\mathbf{i} \in \mathcal{I}$, and $\mathcal{E}$ is the neighboring relationship set for the location index set $\mathcal{I}$. Let p be the cardinality of $\mathcal{E}$, i.e., $p = |\mathcal{E}|$. Denote $\mathbf{g} : \mathbb{R}^{P_1 \times \cdots \times P_D \times K} \to \mathbb{R}^p$ such that $\mathbf{g}(\boldsymbol{\alpha}) = (\|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2)_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}}$. We can then rewrite the non-differentiable fusion term $\sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2$ (the left hand side of (14) of the main paper) as $\|g(\boldsymbol{\alpha})\|_1$. Now, let $\boldsymbol{\xi}$ be a p-dimensional random vector with $\xi_j \overset{\text{i.i.d.}}{\sim} t_2$, $j \in \mathcal{E}$, where t_2 denotes the Student t distribution with 2 degrees of freedom. It can be shown that, for $\epsilon_1 > 0$,

$$\mathbb{E}\,\|g(\boldsymbol{\alpha}) + \sqrt{(\epsilon_1/2)}\,\boldsymbol{\xi}\|_1 = \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \sqrt{\|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2 + \epsilon_1}. \tag{A.10}$$

Thus (14) of the main paper can be represented by a perturbation (Chatterji et al., 2020) using Student t distribution with 2 degrees of freedom. To show (A.10), we note that for a

random variable $\xi \sim t_2$, direct calculation shows

$$\mathbb{E}|\alpha + u\xi| = \sqrt{\alpha + 2u^2}, \tag{A.11}$$

for α, $u \in R$. Thus, for $\xi_{(\mathbf{i},\mathbf{i}')} \overset{\text{i.i.d.}}{\sim} t_2$, we have

$$\begin{aligned}\mathbb{E}\,\|g(\boldsymbol{\alpha}) + \sqrt{(\epsilon_1/2)}\,\boldsymbol{\xi}\|_1 &= \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \mathbb{E}\left|\|\boldsymbol{\alpha}_\mathbf{i} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2 + \sqrt{(\epsilon_1/2)}\,\xi_{(\mathbf{i},\mathbf{i}')}\right| \\ &\overset{(A.11)}{=} \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \sqrt{\|\boldsymbol{\alpha}_\mathbf{i} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2 + \epsilon_1},\end{aligned}$$

which completes our Student t_2 perturbation representation of (14) of the main paper.

Similarly, let U be the indicator function such that $U(u) = \mathbf{1}_{\{u>\lambda\}}$, and define $\xi_\mathbf{i} \overset{\text{i.i.d.}}{\sim} t_1$, $\mathbf{i} \in \mathcal{I}$, where t_1 denotes the Student t distribution with 1 degree of freedom (i.e., the Cauchy distribution). We then have

$$\begin{aligned}\mathbb{E}\big\{U\big(\|\boldsymbol{\alpha}_\mathbf{i}\|_2^2 + \epsilon_0\xi_\mathbf{i}\big)\big\} &= \mathbb{E}\big(\mathbf{1}_{\{\|\boldsymbol{\alpha}_\mathbf{i}\|_2^2+\epsilon_0\xi_\mathbf{i}>\lambda\}}\big) \\ &= \mathbb{P}\big(\|\boldsymbol{\alpha}_\mathbf{i}\|_2^2 + \epsilon_0\xi_\mathbf{i} > \lambda\big) \\ &= \mathbb{P}\big\{\xi_\mathbf{i} > (\lambda - \|\boldsymbol{\alpha}_\mathbf{i}\|_2^2)/\epsilon_0\big\} \\ &= 1 - \left[\frac{1}{2} + \frac{1}{\pi}\arctan\{(\lambda - \|\boldsymbol{\alpha}_\mathbf{i}\|_2^2)/\epsilon_0\}\right] \\ &= \frac{1}{2} + \frac{1}{\pi}\arctan\{(\|\boldsymbol{\alpha}_\mathbf{i}\|_2^2 - \lambda)/\epsilon_0\}.\end{aligned}$$

Thus, (13) of the main paper can be represented by the Student t_1 (Cauchy) perturbation.

A.5 Model Estimation

In this section we demonstrate how to estimate the tensor regression model (1) of the main paper. The method includes two major components: sampling posterior and selecting hyperparameters (a validation method).

A.5.1 Posterior Sampling

Algorithm A.1 describes the Markov chain Monte Carlo (MCMC) method to obtain posterior samples for the parameters $\{\mu, \boldsymbol{\alpha}, \lambda, \sigma^2, \delta\}$ of the hierarchical model (9)–(12) of the main paper. Steps 1 and 2 use the MALA (Roberts and Rosenthal, 1998) to sample μ and $\boldsymbol{\alpha}$; Step 3 and 4 draw the parameters σ^2 and δ from their full conditional probabilities; and Step 5 applies the Metropolis-Hastings algorithm with a truncated normal proposal to update λ (Cai et al., 2020). The hyperparamaters $(r, \rho_{\boldsymbol{\alpha}})$ in (11) and p_0 in (12) of the main paper are fixed during the MCMC update.

Algorithm A.1: Posterior updates under fixed r and $\rho_{\boldsymbol{\alpha}}$.

Input: the parameters from the last iteration

Output: the updated parameters for the next iteration

1 Draw $\mu^* \sim N(\tilde{\mu}, \tau_\mu^2)$, where

$$\tilde{\mu} = \mu + \frac{\tau_\mu^2}{2}\Big(\sum_{n=1}^{N} \frac{\partial \ln p(y_n \mid \boldsymbol{\alpha}, \mu, \sigma^2, \lambda)}{\partial \mu} + \frac{\partial \ln p(\mu)}{\partial \mu}\Big).$$

Update $\mu = \mu^*$ with probability

$$\min\left\{1, \frac{N(\mu|\tilde{\mu^*}, \tau_{\mu^*}^2) p(\mu^*) \prod_{n=1}^{N} p(y_n|\boldsymbol{\alpha}, \mu^*, \sigma^2, \lambda)}{N(\mu^*|\tilde{\mu}, \tau_\mu^2) p(\mu) \prod_{n=1}^{N} p(y_n|\boldsymbol{\alpha}, \mu, \sigma^2, \lambda)}\right\}.$$

2 Draw $\boldsymbol{\alpha}^* \sim N(\tilde{\boldsymbol{\alpha}}, \tau_{\boldsymbol{\alpha}}^2 I_{P_1 P_2})$, where

$$\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \frac{\tau_{\boldsymbol{\alpha}}^2}{2}\Big(\sum_{n=1}^{N} \frac{\partial \ln p(y_n \mid \boldsymbol{\alpha}, \mu, \sigma^2, \lambda)}{\partial \boldsymbol{\alpha}} + \frac{\partial \ln p(\boldsymbol{\alpha} \mid \delta, r, \sigma^2, \rho_{\boldsymbol{\alpha}})}{\partial \boldsymbol{\alpha}}\Big).$$

Update $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$ with probability

$$\min\left\{1, \frac{N(\boldsymbol{\alpha}|\tilde{\boldsymbol{\alpha}^*}, \tau_{\boldsymbol{\alpha}^*}^2 I_{P_1 P_2}) p(\boldsymbol{\alpha}^* \mid \delta, r, \sigma^2, \rho_{\boldsymbol{\alpha}}) \prod_{n=1}^{N} p(y_n|\boldsymbol{\alpha}^*, \mu, \sigma^2, \lambda)}{N(\boldsymbol{\alpha}^*|\tilde{\boldsymbol{\alpha}}, \tau_{\boldsymbol{\alpha}}^2 I_{P_1 P_2}) p(\boldsymbol{\alpha} \mid \delta, r, \sigma^2, \rho_{\boldsymbol{\alpha}}) \prod_{n=1}^{N} p(y_n|\boldsymbol{\alpha}, \mu, \sigma^2, \lambda)}\right\}.$$

3 Draw $\sigma^2 \sim \text{Inv-}\Gamma(a, b)$, where $a = p_1 + \frac{N}{2}$,

$$b = 1 + \frac{\sum_{n=1}^{N}(y_n - \mu - \sum_{\mathbf{i}\in\mathcal{I}} \boldsymbol{\phi}(X_{\mathbf{i}}^{(n)})^{\mathrm{T}} \boldsymbol{\alpha}_{\mathbf{i}} \cdot t_\lambda(\boldsymbol{\alpha}_{\mathbf{i}}))^2}{2} + \delta \sum_{\mathbf{i}\in\mathcal{I}} \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}} \mathbf{R} \boldsymbol{\alpha}_{\mathbf{i}}$$
$$+ \frac{r \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2^2 + (1-r) \sum_{(\mathbf{i},\mathbf{i}')\in\mathcal{E}} \|\boldsymbol{\alpha}_{\mathbf{i}} - \boldsymbol{\alpha}_{\mathbf{i}'}\|_2}{2\rho_{\boldsymbol{\alpha}}}.$$

4 Draw $\delta \sim \Gamma(a, b)$, where $a = p_0$ and

$$b = 1 + \frac{\sum_{\mathbf{i}\in\mathcal{I}} \boldsymbol{\alpha}_{\mathbf{i}}^{\mathrm{T}} \mathbf{R} \boldsymbol{\alpha}_{\mathbf{i}}}{\sigma^2}.$$

5 Draw $\lambda^* \sim N_+(\lambda, 0, \lambda_u, \tau_\lambda^2)$, which is a normal distribution $N(\lambda, \tau_\lambda^2)$ truncated by $[0, \lambda_u]$. Update $\lambda = \lambda^*$ with probability

$$\min\left\{1, \frac{N_+(\lambda|\lambda^*, \lambda_l, \lambda_u, \tau_\lambda^2)\, p(\lambda^*) \prod_{n=1}^{N} f(y_n|\boldsymbol{\alpha}, \mu, \sigma^2, \lambda^*)}{N_+(\lambda^*|\lambda, \lambda_\lambda, \lambda_u, \tau_\lambda^2)\, p(\lambda) \prod_{n=1}^{N} f(y_n|\boldsymbol{\alpha}, \mu, \sigma^2, \lambda)}\right\}.$$

Given the posterior samples of $\boldsymbol{\alpha}, \lambda$ from Algorithm A.1, we estimate the model coefficient $\boldsymbol{\beta}$ in (7) of the main paper in the following way. Denote $\left\{\boldsymbol{\alpha}^{(B+l)}, \lambda^{(B+l)}\right\}_{l=1}^{I-B}$ the posterior samples after burn-in, and D_N the training dataset. We achieve sparsity by selecting the active indices (i, j) from the posterior inclusion probability. The posterior inclusion probability for $\boldsymbol{\beta}_{\mathbf{i}}$ is given by the posterior mean of the indicator function $t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda)$ in (13):

$$\widehat{\Pr}\left(\boldsymbol{\beta}_{\mathbf{i}} \neq 0 \mid D_N\right) = \frac{1}{I-B} \sum_{l=1}^{I-B} t\big(\boldsymbol{\alpha}_{\mathbf{i}}^{(B+l)}; \lambda^{(B+l)}\big).$$

The corresponding additive component function $f_{\mathbf{i}}$, $\mathbf{i} \in \widehat{\mathcal{V}}$, is regarded as active if $\widehat{\Pr}\left(\boldsymbol{\beta}_{\mathbf{i}} \neq 0 \mid D_N\right) > c_0$ for some cut-off value c_0. The estimated active index set is then

$$\widehat{\mathcal{V}}(c_0) = \left\{\mathbf{i} : \widehat{\Pr}\left(\boldsymbol{\beta}_{\mathbf{i}} \neq \mathbf{0} \mid D_N\right) > c_0\right\}.$$

Similar to Hajian-Tilaki (2013), the cut-off value c_0 can be decided according to the receiver operating characteristic (ROC) curve. For this purpose, we introduce several notations. Define two tensors $\mathbf{J}_{\mathcal{V}}, \mathbf{J}_{\widehat{\mathcal{V}}(c_0)} \in \mathbb{R}^{P_1 \times \cdots \times P_D}$ such that

$$(\mathbf{J}_{\mathcal{V}})_{\mathbf{i}} = \begin{cases} 1, & \text{true } \boldsymbol{\beta}_{\mathbf{i}} \neq \mathbf{0}, \\ 0, & \text{otherwise;} \end{cases} \qquad \text{and} \qquad (\mathbf{J}_{\widehat{\mathcal{V}}(c_0)})_{\mathbf{i}} = \begin{cases} 1, & \mathbf{i} \in \widehat{\mathcal{V}}(c_0), \\ 0, & \text{otherwise.} \end{cases}$$

In the above, $\boldsymbol{\beta}_{\mathbf{i}}$ is the true coefficient and thus $\mathbf{J}_{\mathcal{V}}$ can be interpreted as an indicator for the true active index, while $\mathbf{J}_{\widehat{\mathcal{V}}(c_0)}$ can be interpreted as the estimated active index. We also define a tensor $\mathbf{J} \in \mathbb{R}^{P_1 \times \cdots \times P_D}$ whose elements are all ones, i.e., $\mathbf{J}_{\mathbf{i}} = 1$, $\forall \mathbf{i} \in \mathcal{I}$. With these notations, the true negative rate (TNR) and the true positive rate (TPR) for the cut-off value c_0 are respectively defined as

$$\mathrm{TNR}(c_0) = \frac{\langle \mathbf{J} - \mathbf{J}_{\mathcal{V}}, \mathbf{J} - \mathbf{J}_{\widehat{\mathcal{V}}(c_0)} \rangle}{\langle \mathbf{J}, \mathbf{J} - \mathbf{J}_{\mathcal{V}} \rangle}$$

and

$$\mathrm{TPR}(c_0) = \frac{\langle \mathbf{J}_{\mathcal{V}}, \mathbf{J}_{\widehat{\mathcal{V}}(c_0)} \rangle}{\langle \mathbf{J}, \mathbf{J}_{\mathcal{V}} \rangle}.$$

Note that $\mathbf{J}_{\mathcal{V}}$ is unknown, we use the tensor $\mathbf{P} \in \mathbb{R}^{P_1 \times \cdots \times P_D}$ whose element $P_{\mathbf{i}} = \widehat{\Pr}\left(\boldsymbol{\beta}_{\mathbf{i}} \neq \mathbf{0} \mid D_N\right)$ to approximate $\mathbf{J}_{\mathcal{V}}$ in practice. According to Hajian-Tilaki (2013), the estimation of TNR and TPR can be obtained as

$$\widehat{\mathrm{TNR}}(c_0) = \frac{\langle \mathbf{J} - \mathbf{P}, \mathbf{J} - \mathbf{J}_{\widehat{\mathcal{V}}(c_0)} \rangle}{\langle \mathbf{J}, \mathbf{J} - \mathbf{P} \rangle}, \quad \widehat{\mathrm{TPR}}(c_0) = \frac{\langle \mathbf{P}, \mathbf{J}_{\widehat{\mathcal{V}}(c_0)} \rangle}{\langle \mathbf{J}, \mathbf{P} \rangle}.$$

We thus determine the optimal cut-off value c_0 as the one minimizing the distance between the point $(0, 1)$ and the ROC curve, i.e.,

$$\widehat{c}_0 = \mathrm{argmin}_c \sqrt{\left\{1 - \widehat{\mathrm{TPR}}(c)\right\}^2 + \left\{1 - \widehat{\mathrm{TNR}}(c)\right\}^2}.$$

Finally, with the selected $\widehat{c}_0$, the estimated regression coefficient for an active index $\mathbf{i}$ is given by

$$\widehat{\boldsymbol{\beta}}_{\mathbf{i}} = \frac{1}{I-B}\sum_{l=1}^{I-B} \boldsymbol{\alpha}_{\mathbf{i}}^{(B+l)} \cdot t(\boldsymbol{\alpha}_{\mathbf{i}}^{(B+l)}; \lambda^{(B+l)}), \quad \mathbf{i} \in \widehat{\mathcal{V}},$$

and the corresponding estimated additive component function turns to be $\widehat{f}_{\mathbf{i}} = \boldsymbol{\phi}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{\mathbf{i}}$.

A.5.2 The Selection of Approximation Parameter

There are two considerations about the tuning parameter ϵ_0 in the smooth indicator $t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda)$ (13). On one hand, as required by MALA, the indicator should be smooth enough. On the other hand, as an indicator function, its range $[\min_{\mathbf{i}} t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda), \max_{\mathbf{i}} t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda)]$ needs to cover $[0, 1]$ as much as possible. According to Figure A.1, we can see that with a bigger ϵ_0, the

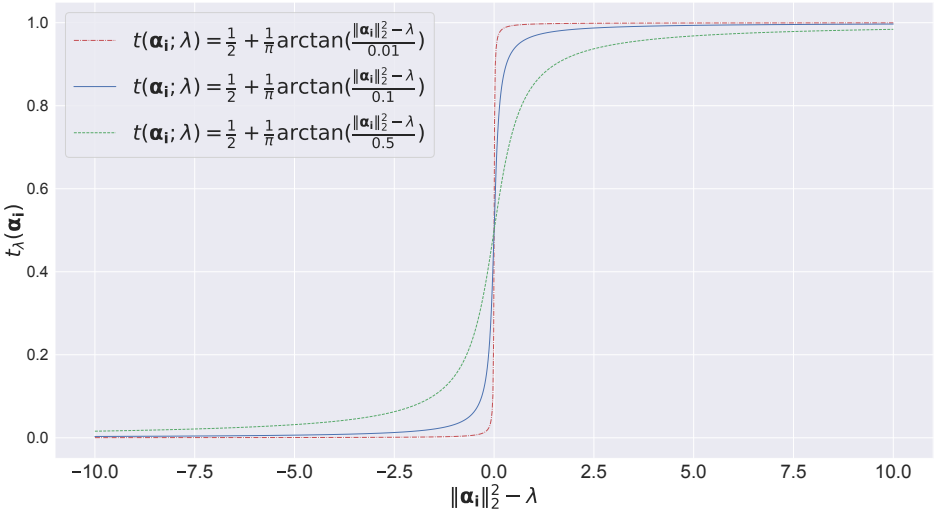

Figure A.1: The smooth indicators with different ϵ_0's.

indicator becomes smoother, but $[\min_{\mathbf{i}} t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda), \max_{\mathbf{i}} t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda)]$ is harder to cover $[0, 1]$. It should be avoided that the parameter ϵ_0 is either too small or too big. Denote m such that $1/2 + (1/\pi)\arctan(m) = 1 - \eta$, where η is close to 0. To make $[\min_{\mathbf{i}} t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda), \max_{\mathbf{i}} t(\boldsymbol{\alpha}_{\mathbf{i}}; \lambda)]$ cover $[\eta, 1-\eta]$, we require ϵ_0 to satisfy $\min_{\mathbf{i}} \|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 - \lambda \leq -m\epsilon_0$ and $\max_{\mathbf{i}} \|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 - \lambda \geq m\epsilon_0$, hence we have

$$\epsilon_0 \in \left(0, \frac{\max_{\mathbf{i}} \|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 - \min_{\mathbf{i}} \|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2}{2m}\right].$$

We choose the largest value

$$\frac{\max_{\mathbf{i}} \|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2 - \min_{\mathbf{i}} \|\boldsymbol{\alpha}_{\mathbf{i}}\|_2^2}{2m}$$

for ϵ_0 to make the indicator smooth as far as possible, and we suggest to set η as 0.05 in the numerical experiments.

However, in a real application, $\|\boldsymbol{\alpha}_\mathbf{i}\|_2^2$ is unknown. In practice, we apply a two-step strategy as follows to settle this problem.

(i) Set $t(\boldsymbol{\alpha}_\mathbf{i}; \lambda) \equiv 1$ and drop the step updating λ in Algorithm A.1. We run Algorithm A.1 under $r = 1$ and $\rho_{\boldsymbol{\alpha}} = \rho_1$ to get rough estimates of $\max_\mathbf{i} \|\boldsymbol{\alpha}_\mathbf{i}\|_2^2$ and $\min_\mathbf{i} \|\boldsymbol{\alpha}_\mathbf{i}\|_2^2$. We then set

$$\epsilon_0 = \frac{\max_\mathbf{i} \|\widehat{\boldsymbol{\alpha}}_\mathbf{i}\|_2^2 - \min_\mathbf{i} \|\widehat{\boldsymbol{\alpha}}_\mathbf{i}\|_2^2}{2m}.$$

(ii) With this ϵ_0, we completely run Algorithm A.1 under $r = 1$ and $\rho_{\boldsymbol{\alpha}} = \rho_1$ to get new estimates of $\max_\mathbf{i} \|\boldsymbol{\alpha}_\mathbf{i}\|_2^2$ and $\min_\mathbf{i} \|\boldsymbol{\alpha}_\mathbf{i}\|_2^2$. We then set

$$\epsilon_0 = \frac{\max_\mathbf{i} \|\widehat{\boldsymbol{\alpha}}_\mathbf{i}\|_2^2 - \min_\mathbf{i} \|\widehat{\boldsymbol{\alpha}}_\mathbf{i}\|_2^2}{2m}.$$

As for the tuning parameter ϵ_1, it is used in the approximation to the ℓ_1 function (see (14) of the main paper). To closely approximate the nonsmooth ℓ_1 function, ϵ_1 is suggested to be small enough and we specify its value to be 10^{-6} as discussed in Section 3.5 of the main paper. We find that our proposed model is not sensitive to the specific value of ϵ_1 through a sensitivity analysis on the simulated data. In particular, we generated the simulated dataset of sample size 600 under the nonlinear setting of a horse shape with high SNR (Setting 5) as in Section 4 of the main paper. On the simulated dataset, we applied our proposed BFEN method with $\epsilon_1 = 10^{-6}$, 10^{-8}, and 10^{-10}. We repeated the experiments 30 times and calculated relative prediction error (RPE), mean squared error (MSE), relative mean squared error (RMSE), true positive rate (TPR), and true negative rate (TNR) as in the main paper. The results for various ϵ_1's are summarized in Table A.1. Table A.1 shows that our model is not sensitive to the tuning parameter ϵ_1 with small values.

Table A.1: Estimation errors for different specifications of parameter ϵ_1. The first row stands for the default specification, i.e., $\epsilon_1 = 10^{-6}$. The following rows summarize results when ϵ_1 is assigned new values. The numbers in parentheses are the standard errors based on 30 replicates.

	RPE	MSE	RMSE	TPR	TNR
$\epsilon_1 = 10^{-6}$	0.0763 (0.0024)	0.0166 (0.0006)	0.0508 (0.0024)	0.9996 (0.0004)	0.9392 (0.0021)
$\epsilon_1 = 10^{-8}$	0.0761 (0.0022)	0.0165 (0.0007)	0.0509 (0.0025)	0.9998 (0.0002)	0.9405 (0.0017)
$\epsilon_1 = 10^{-10}$	0.0763 (0.0022)	0.0168 (0.0008)	0.0516 (0.0028)	0.9996 (0.0004)	0.9412 (0.0017)

Algorithm A.2: Validation method to select p_0 and $\rho_{\boldsymbol{\alpha}}$.

Let $\text{Loss}_{j,t}$ denote the validation loss corresponding to hyperparameters $p_0 = p_{0,j}$ and $\rho_{\boldsymbol{\alpha}} = \rho_{\boldsymbol{\alpha},t}$.

Denote $\boldsymbol{\Theta}^{(j,t,i)} = \{\mu^{(j,t,i)}, \boldsymbol{\alpha}^{(j,t,i)}, (\sigma^2)^{(j,t,i)}, \delta^{(j,t,i)}, \lambda^{(j,t,i)}\}$ the i-th update of posterior samples set under $p_0 = p_{0,j}$ and $\rho_{\boldsymbol{\alpha}} = \rho_{\boldsymbol{\alpha},t}$.

Set $p_{0,1} < \cdots < p_{0,J}, \rho_{\boldsymbol{\alpha},1} < \cdots < \rho_{\boldsymbol{\alpha},T}$, $W < B < I$.

Set $p_0 = p_{0,1}, \rho_{\boldsymbol{\alpha}} = \rho_{\boldsymbol{\alpha},1}$, and initialize the parameter set $\boldsymbol{\Theta}^{(1,1,1)}$.

Obtain $\{\boldsymbol{\Theta}^{(1,1,i)}\}_{i=1}^{I}$ through Algorithm A.1 and set $j = 1, t = 2, j' = 1, t' = 1$.

while $j \leq J$ **do**

  Set $r = r_j, \rho_{\boldsymbol{\alpha}} = \rho_{\boldsymbol{\alpha},t}$.

  Initialize the parameters set $\boldsymbol{\Theta}^{(j,t,W)}$ with the averaged $\{\boldsymbol{\Theta}^{(j',t',i)}\}_{i=B}^{I}$.

  Obtain $\{\boldsymbol{\Theta}^{(j,t,i)}\}_{i=W}^{I}$ through Algorithm A.1, and compute validation loss $\text{Loss}_{j,t}$.

  Set $j' = j, t' = t$.

  if $j \equiv 1 \pmod 2$ **then**

    if $t < T$ **then**

      $t = t + 1$;

    else

      $j = j + 1$

    end

  else

    if $t > 1$ **then**

      $t = t - 1$;

    else

      $j = j + 1$

    end

  end

end

Set $(j_0, t_0) = \text{argmin}_{j,t}\{\text{Loss}_{j,t} \mid 1 \leq j \leq J, 1 \leq t \leq T\}$.

A.5.3 Validation Method

We apply the validation method to select tuning parameters $(r, \rho_{\boldsymbol{\alpha}})$ in prior (11) and p_0 in hyperprior (12) from a corresponding list of candidate values $\{r_s\}_{s=1,\cdots,S}$, $\{\rho_{\boldsymbol{\alpha},t}\}_{t=1,\cdots,T}$, and $\{p_{0,j}\}_{j=1,\cdots,J}$. Given the estimated $\{\widehat{f}_{\mathbf{i}}\}_{\mathbf{i}\in\mathcal{I}}$ from the training set under $p_0 = p_{0,j}$, $r = r_s$, $\rho_{\boldsymbol{\alpha}} = \rho_{\boldsymbol{\alpha},t}$, the response value y is predicted in the validation set. The final tuning parameters are selected as those minimizing the validation loss $\mathrm{L}(\mathbf{y}_{\text{valid}}, \widehat{\mathbf{y}}_{\text{valid}})$.

For the validation method, applying Algorithm A.1 for all combinations of $p_0 \in \{p_{0,j}\}_{j=1,\cdots,J}, r \in \{r_s\}_{s=1,\cdots,S}, \rho_{\boldsymbol{\alpha}} \in \{\rho_{\boldsymbol{\alpha},t}\}_{t=1,\cdots,T}$ is a time-consuming process. We adopt the following strategy to reduce the computational cost.

(a) The number of combinations of $(p_0, r, \rho_{\boldsymbol{\alpha}})$ can be reduced from $J \cdot S \cdot T$ to $J \cdot T + S \cdot T$ by using a two-step greedy search:

   i) Compute the validation loss under different $p_0 \in \{p_{0,j}\}_{j=1,\cdots,J}, \rho_{\boldsymbol{\alpha}} \in \{\rho_{\boldsymbol{\alpha},t}\}_{t=1,\cdots,T}$ with $r = r_1$ fixed. Select $p_0 = p_{0,j_0}$ from the optimal pair $(p_0, \rho_{\boldsymbol{\alpha}})$ that minimizes the validation loss.

   ii) Compute the validation loss under different $r \in \{r_s\}_{s=1,\cdots,S}, \rho_{\boldsymbol{\alpha}} \in \{\rho_{\boldsymbol{\alpha},t}\}_{t=1,\cdots,T}$ with $p_0 = p_{0,j_0}$ fixed, then select the optimal $r = r_{s_0}, \rho_{\boldsymbol{\alpha},t_0}$.

(b) The number of iterations in executing Algorithm A.1 under each $p_0, r, \rho_{\boldsymbol{\alpha}}$ can be reduced by applying a warmstart. In other words, the initial point of Algorithm A.1 under $r = r_2, \rho_{\boldsymbol{\alpha}} = \rho_{\boldsymbol{\alpha},1}$ is determined as the output of Algorithm A.1 under $r = r_1, \rho_{\boldsymbol{\alpha}} = \rho_{\boldsymbol{\alpha},1}$. We find that this initialization trick circularly reduces the computational burden of validation.

The validation method to obtain the optimal tuning parameters p_0 and $\rho_{\boldsymbol{\alpha}}$ are summarized in Algorithm A.2. We omit the detailed algorithm to obtain the optimal tuning parameters r since the procedure is similar. For the candidate grids of p_0, r, and $\rho_{\boldsymbol{\alpha}}$, their ranges should be reasonable and wide enough. Among them, the range of r is within $[0, 1]$ and thus we assign the grid of r as $\{1, 0.75, 0.5, 0.25, 0\}$ following Zhou et al. (2020). For $\rho_{\boldsymbol{\alpha}}$, we specify the grid $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5\}$ for $\rho_{\boldsymbol{\alpha}}$ following the suggestion of Teipel et al. (2015); Engebretsen and Bohlin (2019); Tec et al. (2020). For p_0, its grid is suggested as

$$\{0.5P_1 \cdots P_D K,\ 5P_1 \cdots P_D K,\ 50P_1 \cdots P_D K,\ 500P_1 \cdots P_D K,\ 5000P_1 \cdots P_D K\},$$

where the lower bound of the grid is determined according to Proposition 1 of the main paper. We find all the above grids are wide enough in our numerical experiments.

A.5.4 Sensitivity Analyses

For the other hyperparameters $\{p_1, \sigma_\mu^2, p, a, b\}$ and tuning parameters δ', first note that according to Proposition 1 of the main paper, we set $p_1 = (2P_1 \cdots P_D - 1)K/2$ to balance the magnitude of the normalization term of the prior distribution $p(\delta, \sigma^2)$ in (12) and thus $p(\delta, \sigma^2)$ is weakly informative with respect to σ^2 (similar to an inverse-gamma prior with the small shape parameter). For the rest σ_μ^2, p, a, b, and δ', we carry out a sensitivity analysis for these parameters. Recall that our default specifications are $\sigma_\mu^2 = 100$, $p = 1$, $a = b = 0.5$, and $\delta' = 0.0001$, which renders the prior relatively weak-informative. In particular for a, we plot the mean and the variance as two functions of the parameter a with p and b fixed at 1 and 0.5, respectively. According to Figure A.2, $a = 0.5$ shows reasonably weak-informative since it is at the "elbow" for both curves. In the sensitivity analysis, we consider a larger and a smaller values relative to the default setting for hyperparamters σ_μ^2, p, a, b, and δ' to assess their sensitivity. Results of sensitivity analysis in the nonlinear setting of a horse shape with high SNR (Setting 5) of our simulation experiments are summarized in Table A.2 based on 30 replicates. It can be seen that our model is relatively robust with different choices of tuning/hyper parameters.

Table A.2: Estimation errors for different specifications of tuning/hyper parameters. The first row shows the results with default specification: $\sigma_\mu^2 = 100$, $p = 1$, $a = b = 0.5$, and $\delta' = 0.0001$. The following rows exhibit the results with each parameter being assigned new values. The numbers in parentheses are the standard errors based on 30 replicates.

	RPE	MSE	RMSE	TPR	TNR
default	0.0763 (0.0024)	0.0166 (0.0006)	0.0508 (0.0024)	0.9996 (0.0004)	0.9392 (0.0021)
$\sigma_\mu^2 = 10$	0.0751 (0.0021)	0.0162 (0.0006)	0.0493 (0.0017)	1.0000 (0.0000)	0.9404 (0.0024)
$\sigma_\mu^2 = 1000$	0.0748 (0.0019)	0.0165 (0.0007)	0.0505 (0.0026)	0.9998 (0.0002)	0.9405 (0.0019)
$p = -10$	0.0765 (0.0024)	0.0169 (0.0008)	0.0521 (0.0028)	0.9995 (0.0004)	0.9404 (0.0019)
$p = 10$	0.0757 (0.0020)	0.0166 (0.0007)	0.0514 (0.0022)	1.0000 (0.0000)	0.9427 (0.0023)
$a = 0.25$	0.1002 (0.0026)	0.0235 (0.0016)	0.0701 (0.0048)	0.9991 (0.0004)	0.9237 (0.0050)
$a = 1$	0.0667 (0.0013)	0.0138 (0.0003)	0.0434 (0.0011)	1.0000 (0.0000)	0.9487 (0.0014)
$b = 0.25$	0.0754 (0.0021)	0.0165 (0.0007)	0.0489 (0.0017)	1.0000 (0.0000)	0.9381 (0.0025)
$b = 1$	0.0751 (0.0020)	0.0164 (0.0007)	0.0488 (0.0017)	1.0000 (0.0000)	0.9381 (0.0025)
$\delta' = 0.001$	0.0756 (0.0020)	0.0163 (0.0006)	0.0484 (0.0014)	1.0000 (0.0000)	0.9375 (0.0027)
$\delta' = 0.00001$	0.0772 (0.0023)	0.0167 (0.0007)	0.0514 (0.0020)	1.0000 (0.0000)	0.9420 (0.0025)

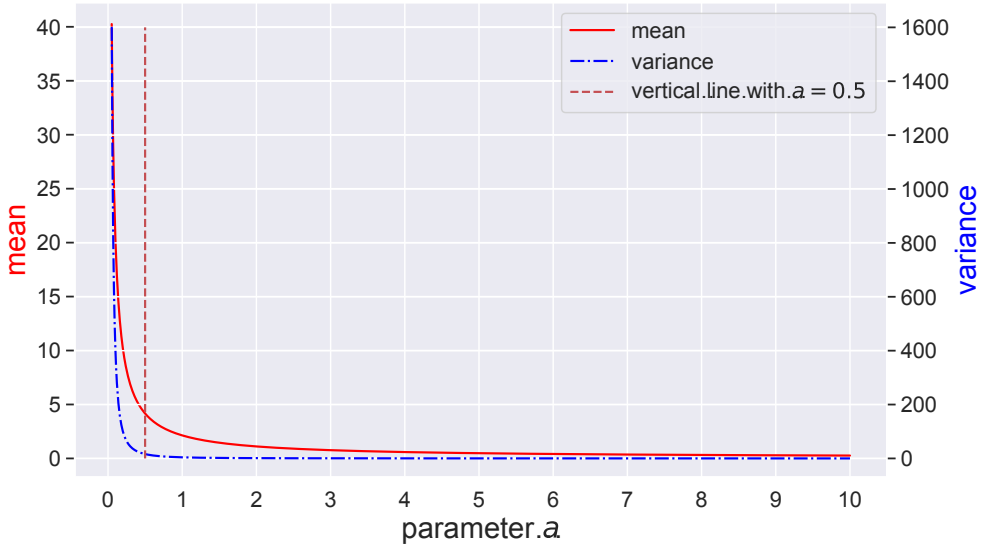

Figure A.2: The mean and the variance of the generalized inverse Gaussian prior (10) as a function of parameter a with p and b fixed as 1 and 0.5, respectively. The plot has a shared x-axis and two y-axes correspondingly for mean (the left) and variance (the right).

A.5.5 Summary

We summarize how to estimate the tensor regression model (1) of the main paper. First, select the hyperparamters ϵ_0 through the methods introduced in Section A.5.2. Second, follow Section A.5.3 to obtain the optimal tuning parameters $(p_0, r, \rho_{\boldsymbol{\alpha}})$. Finally, follow Section A.5.1 to obtain the estimated component functions $f_{\mathbf{i}}$'s through the posterior samples corresponding to the optimal tuning parameters $(p_0, r, \rho_{\boldsymbol{\alpha}})$.

A.6 Tuning Parameter Selection for the Compared Methods

We present here the details of tuning parameter selection for the competitive methods (Zhou et al., 2013; Hao et al., 2021; Guhaniyogi et al., 2017) in our numerical experiments. As suggested in Zhou et al. (2013) and Hao et al. (2021), we choose lasso penalty and group lasso penalty for FTR and STAR respectively, and also apply the validation method to select the rank of CP decomposition and the tuning parameters of their penalties. The training set, validation set, and test set for STAR and FTR are the same as those for BFEN. Note that BTR only needs a training set and a test set because it automatically selects the tuning parameters (Guhaniyogi et al., 2017). So the training set and validation set for BFEN are used together to train BTR.

For FTR and STAR, both of the tuning parameters of lasso (FTR) and group lasso (STAR) are selected from a geometric sequence $(\rho_1, \cdots, \rho_{10})$, where $\rho_1 = 0.1$ and $\rho_{10} = 10$; and the rank of CP decomposition is selected from $\{2, 4, 6, 8, 10\}$. The above grids are wide enough for two competing models in the sense that the boundary points are seldomly selected by either method. For BTR, the hyperparamters are selected as Guhaniyogi et al. (2017) suggested. In particular, the rank of the CP decomposition is selected as 10 in simulation experiments. In real data experiments, we find that BTR fails to converge for some experiments among the 100 replicates if the rank of CP decomposition is over 6. Hence, we select the rank as 5 in real data experiments.

A.7 Additional Results for the Simulation Study

In this section we explain in details how to construct spatially smooth signals through graph Laplacian matrix, and provide some additional outputs for the experimental results.

A.7.1 The Construction of Spatially Smooth Model

The parameters $\{\overline{u}_{ij}^{(1)}\}$, $\{v_{ij}^{(2)}\}$, and $\{v_{ij}^{(3)}\}$ in Table 2 of the main paper are constructed through graph Laplacian matrix as follows.

First, we obtain the graph Laplacian matrix (Merris, 1994), $\mathbf{L} \in \mathbb{R}^{P_1P_2 \times P_1P_2}$, of the graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ defined in Section 2 of the main paper where the index in $\mathcal{I}$ is arranged in the column-major order. Denote $\mathbf{u}_l$ as the eigenvector of $\mathbf{L}$ corresponding to the l-th smallest eigenvalue. We focus on the first L ($L < P_1P_2$) eigenpairs with $l = 1, \ldots, L$. The eigenvector $\mathbf{u}_l \in \mathbb{R}^{P_1P_2}$ is reshaped to be a matrix $\mathbf{U}_l \in \mathbb{R}^{P_1 \times P_2}$ by the column-major order. As the eigenvector correspond to small Laplacian value, the element values of $\mathbf{U}_l$ ($1 \leq l \leq L$) has variability of low frequency and thus is spatially smooth across (i, j) (Dong et al., 2016).

Second, to make use of all the L spatially smooth matrices, we further construct three matrices $\overline{\mathbf{U}}^{(1)}$, $\overline{\mathbf{U}}^{(2)}$, and $\overline{\mathbf{U}}^{(3)}$ as random linear combinations of $\mathbf{U}_1, \cdots, \mathbf{U}_L$, i.e., $\overline{\mathbf{U}}^{(m)} = \sum_{l=1}^{L} \gamma_l^{(m)} \mathbf{U}_l$ with $\gamma_l^{(m)} \overset{\text{i.i.d.}}{\sim} \text{Unif}(0, 1), l = 1, \ldots, L, m = 1, 2, 3$. After that, a_{ij} in non-linear cases of the horse shape (Settings 4 and 5 in Table 2 of the main paper) is set as $1\overline{u}_{ij}^{(1)} + 2$, where $\overline{u}_{ij}^{(1)}$ is the (i, j)-th element of $\overline{\mathbf{U}}^{(1)}$.

As for c_{ij} and d_{ij} in the non-linear cases of a horse shape and a shape of handwritten six (Settings 4, 5, 7, and 8 in Table 2 of the main paper), they are constructed from $\overline{\mathbf{U}}^{(2)}$ and $\overline{\mathbf{U}}^{(3)}$. To be specific, we rescale each element $u_{ij}^{(m)}$ of $\overline{\mathbf{U}}^{(m)}$, $m = 2$ and 3, to get a new matrix $\left(v_{ij}^{(m)}\right)$ such that $\min_{(i,j)\in\mathcal{V}} v_{ij}^{(m)} = \pi$ and $\max_{(i,j)\in\mathcal{V}} v_{ij}^{(m)} = 1.5\pi$, where $\mathcal{V}$ is the set of active pixels. We set c_{ij} and d_{ij} as $v_{ij}^{(2)}$ and $v_{ij}^{(3)}$, respectively. The number L of eigenvectors is set as $L = 80$.

A.7.2 Additional Results

Recall that we have 9 simulation settings with 30 replicates for each setting. We provide the detailed numerical results and runtime for the simulation experiments in Table A.3. All methods were implemented on the same platform with a 2.2-GHz Intel E5-2650 v4 CPU. The results of the experiments under the nonlinear with high SNR settings have already been exhibited in Section 4.2 of the main paper by heatmaps. In this section, we exhibit the results with the median relative prediction error (RPE) under the nonlinear with low SNR and linear settings. In Figures A.3 and A.4, the shade of each square (i, j) indicates the $\mathbb{L}_2$ norm of f_{ij} as is in Figure 5 of the main paper. The heatmaps in the first column exhibits the magnitude $\|f_{ij}\|_{\mathbb{L}_2}$ of the true component function, while in Columns 2–5 exhibit the magnitude $\|\widehat{f}_{ij}\|_{\mathbb{L}_2}$ estimated by BFEN, STAR, FTR, and BTR, respectively. Rows 1–3 correspond to the patterns of low-rank shapes (Settings 1 and 3), a horse shape (Settings 4 and 6) and a shape of handwritten six (Settings 7 and 9), respectively. Figures A.3 and A.4 show that our method outperforms STAR, FTR, and BTR for irregular sparsity shapes, i.e. a horse and a handwritten Arabic six. Moreover, it is also exhibited that all the methods have good performances when signals are linear and the shape of active region is of low rank. These results are consistent with Figures 4 and 5 of the main paper.

A.7.3 A Comparative Study with Random Walk Metropolis

In Algorithm A.1, a hybrid method (Metropolis-adjusted Langevin Algorithm, MALA) and the smoothing technique (Eqn. (13) and Eqn. (14)) are employed for the update of $\boldsymbol{\alpha}$. We chose the MALA over the random walk metropolis due to its computational efficiency. To illustrate this point, we compare MALA and the random walk metropolis under the nonlinear setting of handwritten Arabic six with high SNR (Setting 8) of our simulation experiments. We apply the proposed BFEN model with MALA (Algorithm A.1 of the Appendix) and the random walk metropolis (the corresponding MALA step is replaced by a random walk metropolis step in Algorithm A.1) on the simulated dataset to sample the coefficients $\boldsymbol{\alpha}$ of the unknown functions. Note that since the smoothing technique is no longer involved for the random walk metropolis, ϵ_0 and ϵ_1 are released in this method. The other tuning/hyper parameters of random walk metropolis are set in the same way as the MALA method. In other words, the two methods use the same tuning/hyper parameters except for the extra smoothing parameters in the MALA algorithm. We set the lengths of Markov chains to 20,000 and 50,000 for MALA and random walk metropolis, respectively. In this experiment, the acceptance rate of random walk proposal is around 0.45.

To inspect the convergence of MALA and random walk, we depict the trace plot of average training error $(1/N)\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2$, which is proportional to the negative log-likelihood, of

Table A.3: Average RPE, MSE, RMSE, TPR, TNR, and execution time (in minutes) of various methods in the simulation study. The reported time is the total execution time divided by the number of candidate parameter values in the grid of each method. The numbers in the parentheses are the standard errors based on 30 replicates. The best performances are boldfaced.

Setting ID	1	2	3	4	5	6	7	8	9
Shape	Low rank			Horse			Handwritten Arabic six		
SNR	5	50	5	5	50	5	5	50	5
Setting Meaning	low SNR nonlinear	high SNR nonlinear	linear	low SNR nonlinear	high SNR nonlinear	linear	low SNR nonlinear	high SNR nonlinear	linear
	RPE ($\times 10^{-2}$)								
BFEN	58.49(1.45)	16.87(1.27)	29.79(1.07)	**37.09**(1.01)	**7.63**(0.24)	**23.99**(0.58)	**38.98**(1.30)	**8.39**(1.23)	**20.62**(0.35)
STAR	54.41(1.13)	**12.25**(0.65)	36.61(0.87)	63.52(1.02)	40.38(0.66)	48.10(0.75)	62.75(1.15)	36.10(1.04)	45.65(0.68)
FTR	**46.00**(0.86)	28.01(0.49)	**21.46**(0.39)	63.27(0.72)	49.98(0.69)	36.91(0.48)	54.79(0.87)	39.67(0.70)	29.55(0.48)
BTR	46.74(0.77)	30.01(0.52)	23.76(0.45)	57.96(0.65)	45.36(0.51)	33.41(0.38)	56.19(0.72)	41.75(0.65)	31.26(0.37)
	MSE ($\times 10^{-2}$)								
BFEN	3.86(0.10)	1.15(0.10)	0.18(0.01)	**6.96**(0.27)	**1.66**(0.06)	**0.16**(0.01)	**6.13**(0.28)	**1.51**(0.29)	**0.05**(0.00)
STAR	3.51(0.10)	**0.80**(0.05)	0.26(0.01)	15.82(0.23)	10.92(0.15)	0.66(0.01)	12.75(0.26)	7.87(0.21)	0.35(0.01)
FTR	**2.80**(0.06)	2.09(0.02)	**0.06**(0.00)	15.91(0.16)	13.86(0.13)	0.43(0.01)	10.82(0.12)	8.93(0.09)	0.16(0.00)
BTR	**2.80**(0.03)	2.24(0.02)	0.09(0.00)	14.04(0.09)	12.43(0.08)	0.35(0.00)	10.97(0.11)	9.37(0.07)	0.17(0.00)
	RMSE ($\times 10^{-2}$)								
BFEN	28.43(0.87)	10.34(0.78)	9.26(0.92)	**13.61**(0.79)	**5.08**(0.24)	**6.01**(0.32)	**25.36**(0.76)	**9.14**(1.38)	**3.33**(0.14)
STAR	27.71(1.01)	**5.51**(0.41)	12.32(0.54)	39.58(0.67)	28.51(0.49)	23.69(0.55)	39.02(0.89)	25.45(0.76)	19.18(0.54)
FTR	29.78(0.48)	24.21(0.24)	3.79(0.19)	47.70(0.54)	42.41(0.44)	16.03(0.38)	45.49(0.47)	39.05(0.45)	8.82(0.20)
BTR	**27.27**(0.32)	23.22(0.15)	**2.99**(0.12)	41.67(0.35)	37.19(0.30)	11.12(0.22)	42.17(0.38)	36.60(0.33)	7.90(0.20)
	TPR								
BFEN	0.96(0.01)	**1.00**(0.00)	0.99(0.00)	**0.99**(0.00)	**1.00**(0.00)	**0.99**(0.00)	**0.98**(0.00)	**0.99**(0.00)	**1.00**(0.00)
STAR	**1.00**(0.00)	0.99(0.01)	0.99(0.01)	0.98(0.01)	0.97(0.01)	**0.99**(0.01)	**0.98**(0.01)	0.98(0.01)	0.99(0.01)
FTR	0.96(0.01)	0.99(0.00)	**1.00**(0.00)	0.87(0.01)	0.93(0.01)	0.97(0.00)	0.86(0.01)	0.92(0.01)	0.99(0.00)
BTR	0.97(0.00)	**1.00**(0.00)	**1.00**(0.00)	0.62(0.01)	0.78(0.01)	0.93(0.00)	0.68(0.01)	0.83(0.01)	0.99(0.00)
	TNR								
BFEN	0.78(0.02)	0.91(0.00)	0.91(0.00)	0.81(0.01)	0.94(0.00)	0.93(0.00)	0.85(0.01)	0.93(0.00)	0.94(0.00)
STAR	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)
FTR	0.32(0.02)	0.29(0.02)	0.27(0.02)	0.28(0.03)	0.24(0.03)	0.21(0.02)	0.26(0.02)	0.22(0.02)	0.16(0.02)
BTR	**0.99**(0.00)	**0.98**(0.00)	**0.98**(0.00)	**0.98**(0.00)	**0.97**(0.00)	**0.97**(0.00)	**0.99**(0.00)	**0.98**(0.00)	**0.98**(0.00)
	execution time (in minutes)								
BFEN	1.35(0.02)	1.33(0.02)	1.36(0.02)	1.20(0.02)	1.20(0.02)	1.19(0.01)	1.19(0.04)	1.13(0.01)	1.10(0.01)
STAR	0.57(0.03)	0.86(0.04)	0.78(0.03)	0.46(0.02)	0.57(0.03)	0.59(0.03)	0.39(0.02)	0.49(0.02)	0.49(0.02)
FTR	0.15(0.00)	0.13(0.00)	0.11(0.00)	0.13(0.00)	0.11(0.00)	0.10(0.00)	0.11(0.00)	0.10(0.00)	0.09(0.00)
BTR	17.73(0.12)	17.81(0.15)	17.20(0.27)	14.33(0.11)	14.52(0.14)	14.18(0.02)	13.07(0.01)	12.95(0.04)	12.93(0.04)

both MALA and random walk metropolis. The error is averaged over 10 replicates for the first candidate value of tuning parameters in Figure A.5. Figure A.5 reveals that random walk metropolis fails to explore the posterior efficiently, and that it has not yet converged even with a much longer Markov chain.

We also calculate the relative prediction error (RPE), mean squared error (MSE), relative mean squared error (RMSE), true positive rate (TPR), and true negative rate (TNR). These results are based on the last 1,000 iterations of the two algorithms averaged over 10 replicates, which are summarized in Table A.4. The table also suggests the slow convergence of the

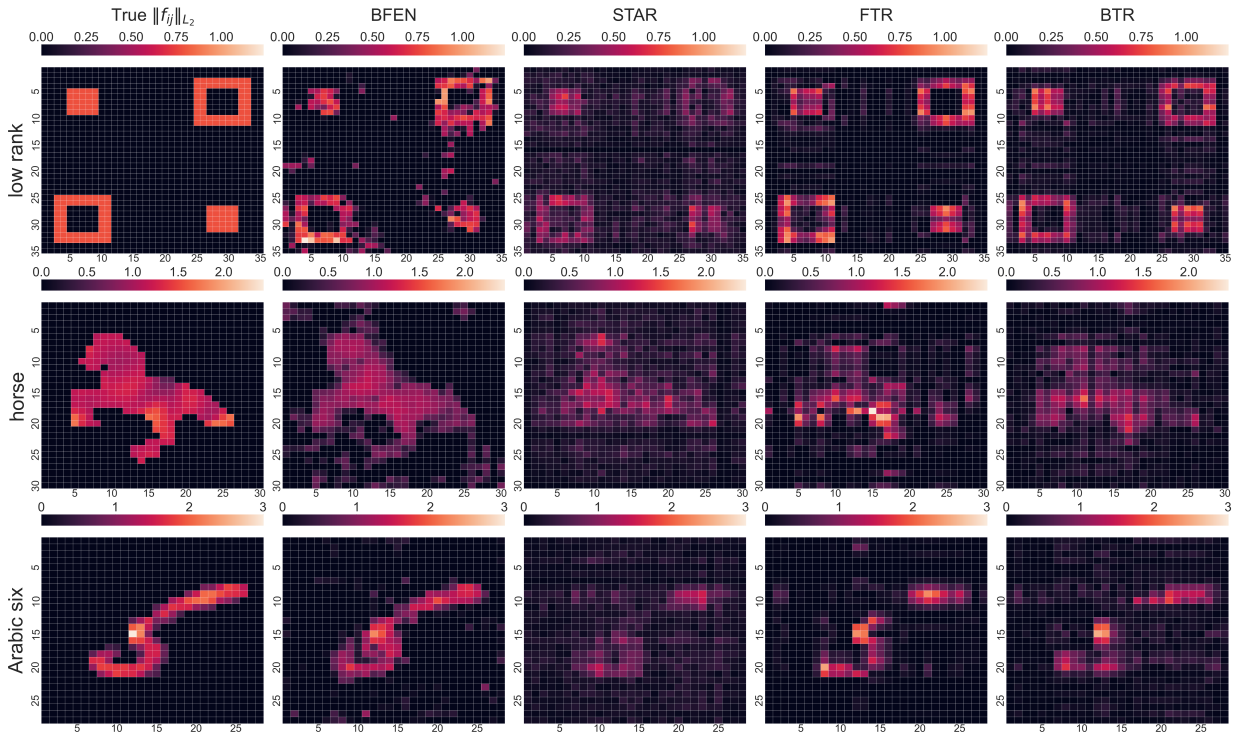

Figure A.3: The heatmaps of various methods under the nonlinear with low SNR settings (Settings 1, 4, and 7). Rows 1–3 correspond to the patterns of low-rank shapes (Setting 1), a horse shape (Setting 4), and a shape of handwritten Arabic six (Setting 7), respectively. The first column presents the magnitude of the true additive component function. Columns 2–5 correspond to the estimated results by BFEN, STAR, FTR, and BTR, respectively.

random walk Metropolis algorithm.

Table A.4: Operating characteristics for MALA and the random walk metropolis. The results are based on 10 replicates.

	RPE	MSE	RMSE	TPR	TNR
MALA	0.08 (0.02)	0.01 (0.00)	0.09 (0.02)	0.99 (0.00)	0.93 (0.00)
random walk	0.38 (0.03)	0.09 (0.01)	0.31 (0.02)	0.90 (0.01)	0.90 (0.01)

A.8 Additional Numerical Results for Facial Data Analysis

We provide the detailed numerical results and runtime for the facial data analysis in Table A.3. All algorithms were run on the same platform with a 2.2-GHz Intel E5-2650 v4 CPU.

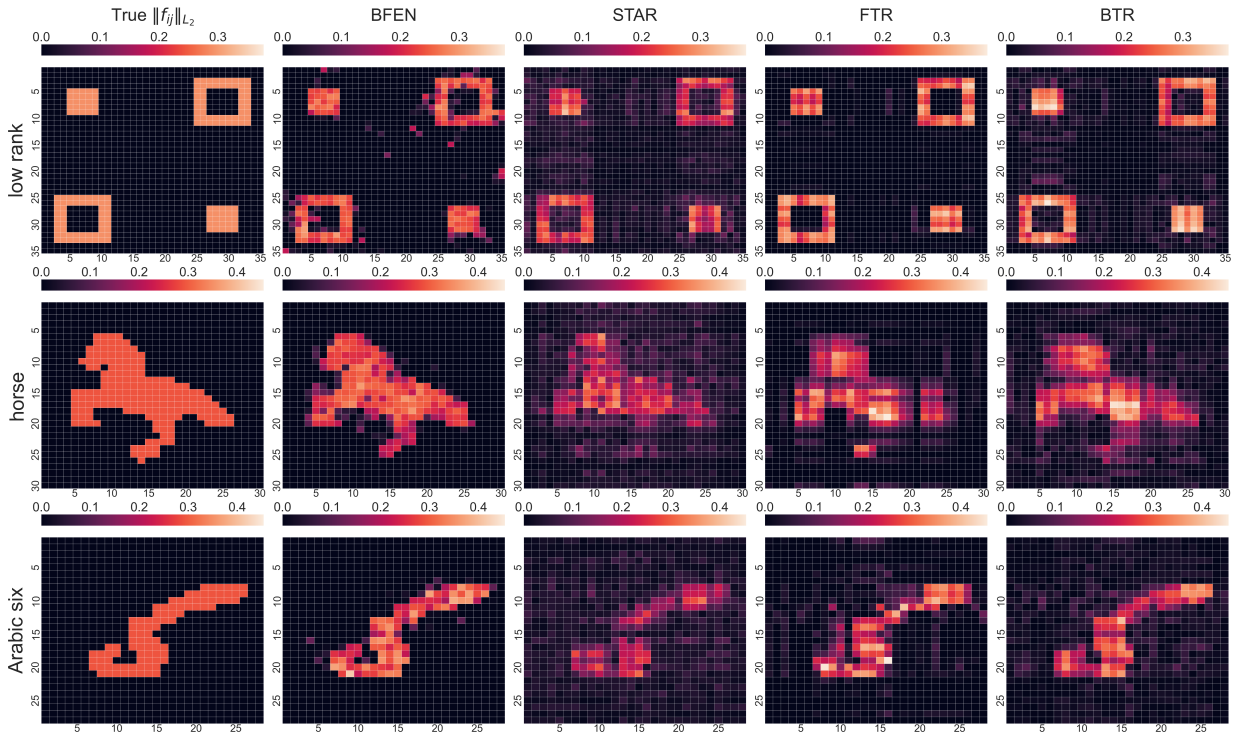

Figure A.4: The heatmaps of various methods under the linear settings (Settings 3, 6, and 9). Rows 1–3 correspond to the patterns of low-rank shapes (Setting 3), a horse shape (Setting 6), and a shape of handwritten Arabic six (Setting 9), respectively. The first column presents the magnitude of the true additive component function. Columns 2–5 correspond to the estimated results by BFEN, STAR, FTR, and BTR, respectively.

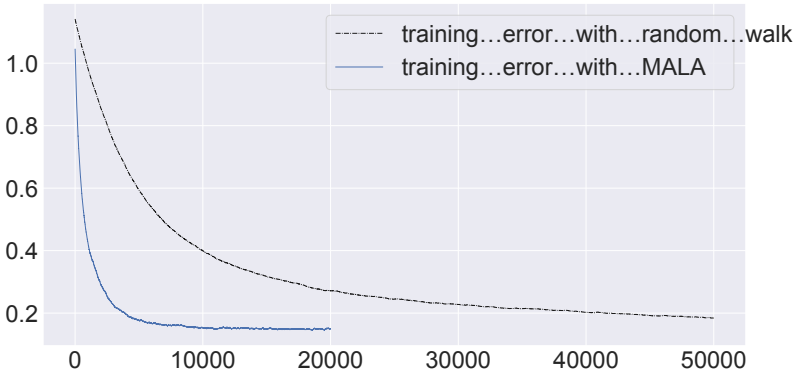

Figure A.5: The trace plot of average training error $(1/N)\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2$ of the Markov chains for MALA and random walk metropolis for the first grid of tuning parameters (i.e., $r = 1$ and $\rho_{\boldsymbol{\alpha}} = 0.001$). The plotted training error at each iteration is the average of 10 replicates. The slight difference of the initial training errors for two algorithms is caused by the extra smoothing approximation employed in MALA.

The magnitude of each estimated additive component functions for the response attribute *smiling* have already been presented in Section 5 of main paper as heatmaps. In this section, we depict the heatmaps of other facial attribute *frowning*, *mouth closed*, *mouth wide open*, and *teeth not visible* in the different rows of Figure A.6. The selected heatmap corresponds to the replicate with the median RPE for each method.

It is evident from the figure that BFEN has better interpretability in most cases. Similar to the attribute *smiling*, the result of *frowning* in the first row of Figure A.6 is also determined by the pixel values around the eyes, mouth and some facial muscles. The attribute *mouth closed* can be determined by the positions of a person's lips and the skin around the lips. When someone keeps his/her *mouth wide open*, the upper lip and lower lip are apart, and thus the mouth cavity can be detected from the image. The teeth is obviously critical for the prediction of the attribute *teeth not visible*. Besides, part of the muscles (orbicularis oris) around the lips are also related to this attribute. In contrast, all the results of FTR and BTR lack interpretations. With the help of nonlinearity and the group regularization across different blocks, STAR has better interpretability than FTR and BTR, but is still inferior to BFEN due to its low-rank modeling. For example, the rectangular subregion selected by STAR may not sufficiently interpret the attribute *mouth wide open*. Overall, our method can achieve a better balance between interpretation and predictive accuracy.

Table A.5: Average RPE and execution time (in minutes) of various methods for each attribute of the facial data analysis. The reported time is the total execution time divided by the number of candidate parameter values in the grid of each method. The numbers in the parentheses are the standard errors based on 100 replicates of random splitting. The best performances are boldfaced.

Attribute	Smiling	Frowning	Mouth closed	Mouth wide open	Teeth not visible
	RPE				
BFEN	**0.2129** (0.0015)	**0.2198** (0.0013)	**0.4510** (0.0025)	**0.2365** (0.0013)	**0.3209** (0.0019)
STAR	0.2233 (0.0014)	0.2314 (0.0015)	0.4647 (0.0027)	**0.2369** (0.0012)	0.3260 (0.0018)
FTR	0.2296 (0.0015)	0.2407 (0.0016)	0.5117 (0.0032)	0.2621 (0.0016)	0.3449 (0.0022)
BTR	0.2501 (0.0026)	0.2599 (0.0024)	0.5136 (0.0042)	0.2641 (0.0024)	0.3748 (0.0038)
	execution time (in minutes)				
BFEN	1.9282 (0.0255)	1.7675 (0.0189)	1.9220 (0.0251)	1.9279 (0.0256)	1.9364 (0.0253)
STAR	2.1964 (0.0254)	2.2022 (0.0241)	1.9066 (0.0265)	2.0286 (0.0400)	2.4710 (0.0686)
FTR	0.7650 (0.0099)	0.7563 (0.0101)	0.4835 (0.0135)	0.4515 (0.0116)	0.6747 (0.0111)
BTR	25.4650 (0.2691)	27.1958 (0.2461)	26.2883 (0.2594)	28.3153 (0.3101)	26.6684 (0.3309)

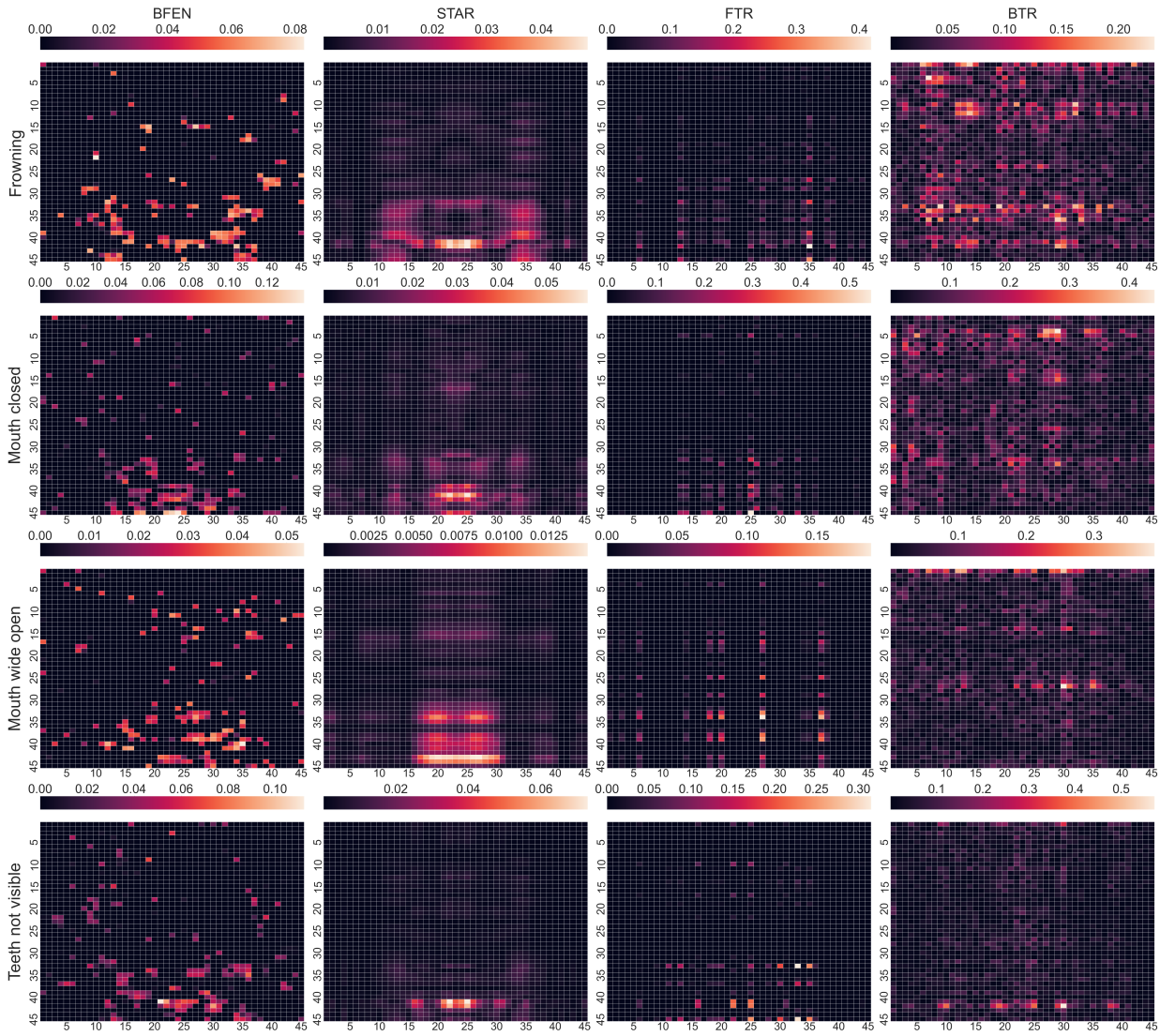

Figure A.6: The heatmaps for the response attributes *frowning*, *mouth closed*, *mouth wide open*, and *teeth not visible*. The shade of square (i, j) in the heatmaps represents the $\mathbb{L}_2$ norm of f_{ij}. The heatmaps in Columns 1–4 correspond to the magnitude $\|\widehat{f}_{ij}\|_{\mathbb{L}_2}$ estimated by BFEN, STAR, FTR, and BTR, respectively.

References

Andersen, C. M. and Bro, R. "Practical aspects of PARAFAC modeling of fluorescence excitation-emission data." Journal of Chemometrics: A Journal of the Chemometrics Society, 17(4):200–215 (2003).

Andrews, D. F. and Mallows, C. L. "Scale mixtures of normal distributions." Journal of the Royal Statistical Society: Series B (Methodological), 36(1):99–102 (1974).

Beer, J. C., Aizenstein, H. J., Anderson, S. J., and Krafty, R. T. "Incorporating prior information with fused sparse group lasso: Application to prediction of clinical measures from neuroimages." Biometrics, 75(4):1299–1309 (2019).

Bohte, S. M., Kok, J. N., and La Poutré, J. A. "SpikeProp: Backpropagation for networks of spiking neurons." In ESANN, volume 48, 419–424. Bruges (2000).

Cai, Q., Kang, J., and Yu, T. "Bayesian network marker selection via the thresholded graph Laplacian Gaussian prior." Bayesian Analysis, 15(1):79–102 (2020).

Chatterji, N., Diakonikolas, J., Jordan, M. I., and Bartlett, P. "Langevin monte carlo without smoothness." In International Conference on Artificial Intelligence and Statistics, 1716–1726. PMLR (2020).

Chen, M.-H. and Shao, Q.-M. "Monte Carlo estimation of Bayesian credible and HPD intervals." Journal of Computational and Graphical Statistics, 8(1):69–92 (1999).

Chew, P. A., Bader, B. W., Kolda, T. G., and Abdelali, A. "Cross-language information retrieval using PARAFAC2." In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 143–152 (2007).

Collins, A. and Koechlin, E. "Reasoning, learning, and creativity: Frontal lobe function and human decision-making." PLoS Biology, 10(3):e1001293 (2012).

Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P. "Learning Laplacian matrix in smooth graph signal representations." IEEE Transactions on Signal Processing, 64(23):6160–6173 (2016).

Ecker, C., Ronan, L., Feng, Y., Daly, E., Murphy, C., Ginestet, C. E., Brammer, M., Fletcher, P. C., Bullmore, E. T., Suckling, J., et al. "Intrinsic gray-matter connectivity of the brain in adults with autism spectrum disorder." Proceedings of the National Academy of Sciences, 110(32):13222–13227 (2013).

Engebretsen, S. and Bohlin, J. "Statistical predictions with glmnet." Clinical epigenetics, 11(1):1–3 (2019).

Fan, J., Feng, Y., and Song, R. "Nonparametric independence screening in sparse ultra-high-dimensional additive models." Journal of the American Statistical Association, 106(494):544–557 (2011).

Fang, X., Paynabar, K., and Gebraeel, N. "Image-based prognostics using penalized tensor regression." Technometrics, 61(3):369–384 (2019).

Fiot, J.-B., Raguet, H., Risser, L., Cohen, L. D., Fripp, J., Vialard, F.-X., Initiative, A. D. N., et al. "Longitudinal deformation models, spatial regularizations and learning strategies to quantify Alzheimer's disease progression." NeuroImage: Clinical, 4:718–729 (2014).

Goldsmith, J., Huang, L., and Crainiceanu, C. M. "Smooth scalar-on-image regression via spatial Bayesian variable selection." Journal of Computational and Graphical Statistics, 23(1):46–64 (2014).

Grill-Spector, K. and Malach, R. "The human visual cortex." Annu. Rev. Neurosci., 27:649–677 (2004).

Grinberg, D. "Notes on the combinatorial fundamentals of algebra." arXiv preprint arXiv:2008.09862 (2020).

Guhaniyogi, R., Qamar, S., and Dunson, D. B. "Bayesian tensor regression." Journal of Machine Learning Research, 18(79):1–31 (2017).

Hajian-Tilaki, K. "Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation." Caspian journal of internal medicine, 4(2):627 (2013).

Hao, B., Wang, B., Wang, P., Zhang, J., Yang, J., and Sun, W. W. "Sparse tensor additive regression." Journal of Machine Learning Research, 22(64):1–43 (2021).

Harshman, R. "Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-mode factor analysis." UCLA Working Papers in Phonetics, 16:1–84 (1970).

Hassner, T., Harel, S., Paz, E., and Enbar, R. "Effective face frontalization in unconstrained images." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4295–4304 (2015).

Hoerl, A. E. and Kennard, R. W. "Ridge regression: Biased estimation for nonorthogonal problems." Technometrics, 12(1):55–67 (1970).

Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments." In Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008).

Huang, J., Horowitz, J. L., and Wei, F. "Variable selection in nonparametric additive models." Annals of statistics, 38(4):2282 (2010).

Imaizumi, M. and Hayashi, K. "Doubly decomposing nonparametric tensor regression." In International Conference on Machine Learning, 727–736. PMLR (2016).

Kanagawa, H., Suzuki, T., Kobayashi, H., Shimizu, N., and Tagami, Y. "Gaussian process nonparametric tensor estimator and its minimax optimality." In International Conference on Machine Learning, 1632–1641. PMLR (2016).

Kandel, B. M., Wolk, D. A., Gee, J. C., and Avants, B. "Predicting cognitive data from medical images using sparse linear regression." In International Conference on Information Processing in Medical Imaging, 86–97. Springer (2013).

Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. "Attribute and simile classifiers for face verification." In 2009 IEEE 12th International Conference on Computer Vision, 365–372. IEEE (2009).

LeCun, Y. "The MNIST database of handwritten digits." http://yann. lecun. com/exdb/mnist/ (1998).

Li, F., Zhang, T., Wang, Q., Gonzalez, M. Z., Maresh, E. L., and Coan, J. A. "Spatial Bayesian variable selection and grouping for high-dimensional scalar-on-image regression." The Annals of Applied Statistics, 9(2):687–713 (2015).

Li, Q., Chen, Y., Jiang, L. L., Li, P., and Chen, H. "A tensor-based information framework for predicting the stock market." ACM Transactions on Information Systems (TOIS), 34(2):1–30 (2016).

Li, X., Xu, D., Zhou, H., and Li, L. "Tucker tensor regression and neuroimaging analysis." Statistics in Biosciences, 10(3):520–545 (2018).

Little, M. A. and Jones, N. S. "Sparse Bayesian step-filtering for high-throughput analysis of molecular machine dynamics." In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 4162–4165. IEEE (2010).

Marx, B. D., Eilers, P. H., and Li, B. "Multidimensional single-index signal regression." Chemometrics and Intelligent Laboratory Systems, 109(2):120–130 (2011).

Merris, R. "Laplacian matrices of graphs: A survey." Linear Algebra and Its Applications, 197:143–176 (1994).

Miao, H., Wang, A., Li, B., and Shi, J. "Structural tensor-on-tensor regression with interaction effects and its application to a hot rolling process." Journal of Quality Technology, 1–14 (2021).

Michel, V., Gramfort, A., Varoquaux, G., Eger, E., and Thirion, B. "Total variation regularization for fMRI-based prediction of behavior." IEEE Transactions on Medical Imaging, 30(7):1328–1340 (2011).

Mitchell, T. J. and Beauchamp, J. J. "Bayesian variable selection in linear regression." Journal of the American Statistical Association, 83(404):1023–1032 (1988).

Ni, Y., Stingo, F. C., and Baladandayuthapani, V. "Bayesian graphical regression." Journal of the American Statistical Association, 114(525):184–197 (2019).

Onaran, I., Ince, N. F., and Cetin, A. E. "Sparse spatial filter via a novel objective function minimization with smooth ℓ_1 regularization." Biomedical Signal Processing and Control, 8(3):282–288 (2013).

Park, S.-T. and Chu, W. "Pairwise preference regression for cold-start recommendation." In Proceedings of the Third ACM Conference on Recommender Systems, 21–28 (2009).

Rischard, M., Pillai, N., and McKinnon, K. A. "Bias correction in daily maximum and minimum temperature measurements through Gaussian process modeling." arXiv preprint arXiv:1805.10214 (2018).

Roberts, G. O. and Rosenthal, J. S. "Optimal scaling of discrete approximations to Langevin diffusions." Journal of the Royal Statistical Society: Series B (Statistical Methodology), 60(1):255–268 (1998).

Rue, H. and Held, L. Gaussian Markov random fields: Theory and applications. Chapman and Hall/CRC (2005).

Ruppert, D., Wand, M. P., and Carroll, R. J. Semiparametric regression. Cambridge: Cambridge University Press (2003).

Shi, J. "In-process quality improvement: Concepts, methodologies, and applications." IISE transactions, 55(1):2–21 (2023).

Signoretto, M., De Lathauwer, L., and Suykens, J. A. "Learning tensors in reproducing kernel Hilbert spaces with multilinear spectral penalties." arXiv preprint arXiv:1310.4977 (2013).

Stone, C. J. "Additive regression and other nonparametric models." The Annals of Statistics, 13(2):689–705 (1985).

Tec, M., Zuniga-Garcia, N., Machemehl, R. B., and Scott, J. G. "Large-Scale Spatiotemporal Density Smoothing with the Graph-fused Elastic Net: Application to Ride-sourcing Driver Productivity Analysis." arXiv preprint arXiv:1911.08106 (2019).

—. "How likely are ride-share drivers to earn a living wage? large-scale spatio-temporal density smoothing with the graph-fused elastic net." arXiv preprint arXiv:1911.08106v2 (2020).

Teipel, S. J., Kurth, J., Krause, B., Grothe, M. J., Initiative, A. D. N., et al. "The relative importance of imaging markers for the prediction of Alzheimer's disease dementia in mild cognitive impairment—beyond classical regression." NeuroImage: Clinical, 8:583–593 (2015).

Tibshirani, R. "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288 (1996).

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. "Sparsity and smoothness via the fused lasso." Journal of The Royal Statistical Society Series B-statistical Methodology, 67:91–108 (2005).

Tucker, L. R. "Some mathematical notes on three-mode factor analysis." Psychometrika, 31(3):279–311 (1966).

Wang, X., Zhu, H., and Initiative, A. D. N. "Generalized scalar-on-image regression models via total variation." Journal of the American Statistical Association, 112(519):1156–1168 (2017).

Xin, B., Kawahara, Y., Wang, Y., and Gao, W. "Efficient generalized fused lasso and its application to the diagnosis of Alzheimer's disease." In Proceedings of the AAAI Conference on Artificial Intelligence, volume 28 (2014).

Yan, H., Paynabar, K., and Pacella, M. "Structured point cloud data analysis via regularized tensor regression for process modeling and optimization." Technometrics, 61(3):385–395 (2019).

Yue, X., Park, J. G., Liang, Z., and Shi, J. "Tensor mixed effects model with application to nanomanufacturing inspection." Technometrics, 62(1):116–129 (2020).

Zhao, Q., Zhou, G., Adali, T., Zhang, L., and Cichocki, A. "Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data." IEEE Signal Processing Magazine, 30(4):137–148 (2013).

Zhao, Q., Zhou, G., Zhang, L., and Cichocki, A. "Tensor-variate gaussian processes regression and its application to video surveillance." In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1265–1269. IEEE (2014).

Zhong, Z., Paynabar, K., and Shi, J. "Image-based feedback control using tensor analysis." Technometrics, (just-accepted):1–14 (2022).

Zhou, H., Li, L., and Zhu, H. "Tensor regression with applications in neuroimaging data analysis." Journal of the American Statistical Association, 108(502):540–552 (2013).

Zhou, Y., Wong, R. K., and He, K. "Broadcasted nonparametric tensor regression." arXiv preprint arXiv:2008.12927 (2020).