Statistica Sinica Preprint No: SS-2024-0402	
Title	Robust Mean Signal Estimation and Inference for Imaging Data
Manuscript ID	SS-2024-0402
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202024.0402
Complete List of Authors	Yang Long, Guanqun Cao, David Kepplinger and Lily Wang
Corresponding Authors	Lily Wang
E-mails	lwang41@gmu.edu
Notice: Accepted author version.	

Robust Mean Signal Estimation and Inference for Imaging Data

Yang LONG[1], Guanqun CAO[2], David KEPPLINGER[1], and Lily WANG[1]

[1] *George Mason University, Fairfax, VA 22030, USA.*

[2]*Michigan State University, East Lansing, MI 48824, USA.*

Abstract: In this paper, we propose a robust nonparametric method for learning and inferring from noisy imaging data by modeling it as contaminated functional data, which enables accurate estimation of the underlying signals and efficient detection and localization of significant effects. The proposed robust and smoothed M-estimator is based on bivariate penalized splines over triangulation, effectively addressing challenges posed by contaminated imaging data, spatial dependencies, and irregular domains commonly encountered in imaging applications. We establish the L_2 convergence of the proposed M-based mean function estimator under certain regularity conditions and investigate its asymptotic normality. Additionally, we present a novel approach for constructing a simultaneous confidence corridor for the mean signal of a set of noisy imaging data. Extensive simulation studies and a real-data application with brain imaging data illustrate the effectiveness of the proposed robust methods.

Key words and phrases: Complex domain; Penalized Splines; Robust estimation; Robust Inference; Simultaneous Confidence Corridors; Triangulation.

1. Introduction

Recent technological advances in imaging hardware, computational power, and data storage have significantly facilitated the collection and analysis of complex datasets, particularly in the field of bioimaging. High-resolution imaging modalities, such as functional magnetic resonance imaging, positron emission tomography (PET), computed tomography, and advanced microscopy, now generate unprecedented volumes of detailed biological data. Inherently, such imaging data are susceptible to multifaceted sources of noise, including acquisition methodology artifacts, environmental perturbations, and subject-induced movement, which can systematically distort image representations and introduce substantial uncertainties in downstream analytical processes. Many imaging data applications now involve machine

learning and deep learning models, which can be highly sensitive to noisy data and outliers. Consequently, developing analytical methodologies that demonstrate robust statistical resilience and adaptive performance is crucial. By implementing rigorous statistical approaches capable of accurately estimating signal components and efficiently detecting and localizing statistical effects, researchers can substantially mitigate the deleterious impacts of data variability and extract more precise and reproducible insights from complex imaging datasets.

Functional Data Analysis (FDA) offers sophisticated statistical methods tailored to imaging analysis (Zhu et al., 2014; Wang et al., 2020; Huang and Zhu, 2022; Shieh and Ogden, 2023). By conceptualizing imaging data as random functions on a bounded two-dimensional (2D) domain, FDA enables a more nuanced interpretation of complex spatial relationships. Given the inherent variability and noise characteristics of imaging data, robust FDA techniques are essential to mitigate potential statistical distortions and ensure reliable analytical results and inference. Recent years have seen a variety of robust FDA techniques. For instance, Gervini (2008) developed a robust estimator of the location parameter of contaminated datasets, as well as a robust functional principal component analysis (FPCA) based on the spherical principal components. Robust FPCA is also studied in Bali et al. (2011); Lee et al. (2013); Boente and Salibian-Barrera (2015). Moreover, robust functional linear regression has been studied in Shin and Lee (2016); Kalogridis and Van Aelst (2023b). For the mean function estimation problem in FDA, Lima et al. (2019a,b) developed robust B-spline smoothed estimators of the mean function for densely observed functional data. An M-type smoothing spline estimator of the mean function based on discretely sampled functional data is also studied in Kalogridis and Van Aelst (2023a). While previous works primarily focused on one-dimensional (1D) curve data, emerging research has begun exploring robust mean function estimation for 2D imaging data using advanced computational techniques such as deep neural networks (Wang and Cao, 2022).

In this paper, we model imaging data as 2D functional data, treating complex imaging

datasets as realizations of random functions. We propose a robust nonparametric method for estimating the mean signal from imaging data by modeling it as contaminated functional data, enabling accurate signal reconstruction and mitigating the impact of noise and outliers. To approximate the mean function, we employ bivariate splines over triangulations (Lai and Schumaker, 2007; Lai and Wang, 2013), leveraging their flexibility, varying amounts of smoothness, and capability of handling irregular domains, which are commonly seen in biomedical imaging data.

Simultaneous inference serves as a fundamental tool for understanding the global behavior of functions, with extensive work in FDA settings (Gu et al., 2014; Degras, 2017; Chang et al., 2017; Choi and Reimherr, 2017; Wang et al., 2020; Li and Yang, 2023; Zhong and Yang, 2023). Existing research predominantly focuses on clean curve data, with limited exploration of robust inference under noisy conditions. Lima et al. (2019a,b) constructed simultaneous confidence corridors (SCCs; also known as simultaneous confidence bands or simultaneous confidence envelopes) using pseudo-data and robust functional principal component analysis. Zhou et al. (2023) proposed a functional response quantile regression model, introducing a global test for coefficient functions and discussing SCC construction. While Wang et al. (2020) examined the SCCs for the mean signal/functions of the imaging data, our simulation results (Section 6) reveal their vulnerability to outliers. To the best of our knowledge, robust SCCs for multi-dimensional functional and imaging data remain unexplored. In this paper, we introduce a comprehensive framework for robust statistical inference in the FDA context and develop a weighted bootstrap method to obtain robust SCCs for the mean function.

The proposed robust learning and inference approach offers three notable contributions to imaging data analysis. First, it is the first method addressing robust learning for imaging data with complex, non-standard anatomical shapes, such as those encountered in brain, heart, and other organ imaging, which deviate from the simplified rectangular representation typical of existing statistical methods. Second, our approach provides both pointwise and

simultaneous robust inference, a novel methodological advancement that extends beyond current robust inference methods restricted to pointwise analysis. This capability enables a more comprehensive characterization of global structural features, which is particularly critical in medical imaging research. Finally, compared to existing robust deep learning techniques, our approach is not only computationally efficient but also theoretically sound.

The remaining sections of this paper are organized as follows. We present the model and estimation method in Section 2, followed by an analysis of its asymptotic behavior in Section 3. We present how to construct robust SCCs in Section 4. In Section 5, we discuss the implementation details for estimation and inference. The finite sample performance of this robust estimation and inference approach is evaluated through a simulation study in Section 6. Section 7 demonstrates an application to an Alzheimer's disease neuroimaging initiative (ADNI) dataset, and we conclude the paper in Section 8. The supplementary materials contain additional simulation results and technical proofs.

2. Models and Estimation Methods

We consider the following functional mean regression model:

$$Y_i(\boldsymbol{z}) = \mu(\boldsymbol{z}) + e_i(\boldsymbol{z}),\ i = 1, \ldots, n,\ \boldsymbol{z} = (z_1, z_2) \in \Omega \subset \mathbb{R}^2, \tag{2.1}$$

where $Y_i(\boldsymbol{z})$ denotes the imaging measurement at location $\boldsymbol{z}$ for the ith subject, $\mu(\boldsymbol{z})$ represents the unknown mean function evaluated at location $\boldsymbol{z}$, and $\{e_i(\cdot)\}_{i=1}^n$ are independent random noise. In practice, the imaging response variable $Y_i(\cdot)$ is measured on a grid of pixel locations $\boldsymbol{z}_j \in \Omega$, $j = 1, \ldots, N$. Let $Y_{ij} = Y_i(\boldsymbol{z}_j)$ be the observed value of the ith subject at pixel location $\boldsymbol{z}_j$. The model (2.1) can be written as $Y_{ij} = \mu(\boldsymbol{z}_j) + e_{ij}$, where $e_{ij} = e_i(\boldsymbol{z}_j)$.

2.1. Bivariate Penalized Spline Over Triangulation

The domain Ω is first partitioned into a finite number of triangles. A collection $\triangle = \{\tau_1, \ldots, \tau_M\}$ of M triangles is called a triangulation of $\Omega = \bigcup_{t=1}^M \tau_t$, provided that any nonempty intersection between a pair of triangles in $\triangle$ is either a shared vertex or a shared

edge. Given a triangle $\tau \in \triangle$, let $|\tau|$ be the length of its longest edge, and the size of $\triangle$ is defined as $|\triangle| := \max\{|\tau|, \tau \in \triangle\}$, which is the length of the longest edge among all triangles in $\triangle$.

For any triangle $\tau \in \triangle$ and any fixed point $\boldsymbol{z} \in \Omega$, let $b_1,\ b_2$, and b_3 be the barycentric coordinates of $\boldsymbol{z}$ relative to τ. Then the Bernstein basis polynomials of degree $d \geq 1$ relative to triangle τ are defined as $B_{ijk}^{\tau,d}(\boldsymbol{z}) = \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k,\ i+j+k=d$. Let $\mathbb{P}_d(\tau)$ be the space of all polynomials of a degree less than or equal to d on τ. Then, any polynomial $\zeta \in \mathbb{P}_d(\tau)$ can be written as $\zeta|_\tau = \sum_{i+j+k=d} \gamma_{ijk}^\tau B_{ijk}^{\tau,d}$, where γ_{ijk}^τ are the coefficients.

For any integer $r \geq 0$, let $\mathbb{C}^r(\Omega)$ be the collection of all r-th order continuously differentiable functions over Ω. Given triangulation $\triangle$, define $\mathbb{S}_d^r(\triangle) = \{\zeta \in \mathbb{C}^r(\Omega) : \zeta|_\tau \in \mathbb{P}_d(\tau), \tau \in \triangle\}$ to be the spline space of degree d and smoothness r over $\triangle$. Let $\{B_m\}_{m\in\mathcal{M}}$ be the set of bivariate Bernstein basis polynomials for $\mathbb{S}_d^r(\triangle)$, where $\mathcal{M}$ is an index set for $|\mathcal{M}| = M(d+1)(d+2)/2$ basis functions, and let $\mathbf{B}$ be the design matrix of Bernstein basis polynomials, where the j-th row is $\mathbf{B}^\top(\boldsymbol{z}_j) = \{B_m(\boldsymbol{z}_j), m \in \mathcal{M}\}$, $j = 1, \ldots, N$.

2.2. Bivariate Penalized Spline Over Triangulation and Smoothed M-Estimator

M-estimation is introduced in Huber (1964), and the regression M-estimator is discussed in Huber (1973). A common choice for the regression loss function in M-estimation is the family of Huber functions:

$$\rho(x;k) = x^2/2\, \mathbb{I}(|x| \leq k) + \left(k|x| - k^2/2\right)\, \mathbb{I}(|x| > k), \tag{2.2}$$

where $\mathbb{I}(\cdot)$ is the indicator function and $k > 0$ is a tuning parameter such that residuals larger than k have a reduced effect on the regression estimation. Note that $k \to \infty$ corresponds to the ordinary least-squares-based estimator, whereas $k = 0$ corresponds to the L_1 estimator.

For any function $g(\boldsymbol{z})$, let $\nabla_{z_s}^l g(\boldsymbol{z})$ be the l-th order derivative in the direction z_s, $s = 1, 2$, at the point $\boldsymbol{z} = (z_1, z_2)$. We consider the following M-type penalized spline regression problem: $\min_{g\in\mathbb{S}_d^r(\triangle)} \sum_{i=1}^n \sum_{j=1}^N \rho\left\{Y_{ij} - g(\boldsymbol{z}_j)\right\} + (h/2)\mathcal{E}(g)$, where $h > 0$ is

the roughness penalty parameter, and $\mathcal{E}(g)$ is the roughness penalty defined as $\mathcal{E}(g) = \int_\Omega (\nabla^2_{z_1} g)^2 + 2(\nabla_{z_1}\nabla_{z_2} g)^2 + (\nabla^2_{z_2} g)^2 dz_1 dz_2$. To meet the smoothness requirements of the splines, some linear constraints on the spline coefficients $\boldsymbol{\gamma}$ have to be imposed, namely $\mathbf{H}\boldsymbol{\gamma} = \mathbf{0}$. In this way, the minimization problem of approximating the mean function using the bivariate spline basis functions can then be expressed as

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{M}|}} \sum_{i=1}^{n}\sum_{j=1}^{N} \rho\left\{Y_{ij} - \mathbf{B}^\top(\boldsymbol{z}_j)\boldsymbol{\gamma}\right\} + \frac{h}{2}\boldsymbol{\gamma}^\top \mathbf{P}\boldsymbol{\gamma} \quad \text{subject to } \mathbf{H}\boldsymbol{\gamma} = \mathbf{0}. \tag{2.3}$$

To solve the constrained minimization problem (2.3), we first remove the constraint by using the QR decomposition: $\mathbf{H}^\top = \mathbf{Q}\mathbf{R} = [\,\mathbf{Q}_1\ \mathbf{Q}_2\,]\left[\begin{smallmatrix}\mathbf{R}_1\\ \mathbf{R}_2\end{smallmatrix}\right]$, where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}_1$ is an upper triangle matrix. The submatrix $\mathbf{Q}_1$ represents the first p columns of $\mathbf{Q}$, where p is the rank of $\mathbf{H}$, and $\mathbf{R}_2$ is a matrix of zeros. Next, we reparameterize using $\boldsymbol{\gamma} = \mathbf{Q}_2\boldsymbol{\theta}$ for some $\boldsymbol{\theta}$, which guarantees that $\mathbf{H}\boldsymbol{\gamma} = \mathbf{0}$. Then, the minimization problem (2.3) is converted to an unconstrained one,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{|\widetilde{\mathcal{M}}|}} \sum_{i=1}^{n}\sum_{j=1}^{N} \rho\left\{Y_{ij} - \widetilde{\mathbf{B}}^\top(\boldsymbol{z}_j)\boldsymbol{\theta}\right\} + \frac{h}{2}\boldsymbol{\theta}^\top \mathbf{D}\boldsymbol{\theta}, \tag{2.4}$$

where $\widetilde{\mathbf{B}} = \mathbf{B}\mathbf{Q}_2$ with rows $\widetilde{\mathbf{B}}^\top(\boldsymbol{z}_j) = \{\widetilde{B}_{\widetilde{m}}(\boldsymbol{z}_j), \widetilde{m} \in \widetilde{\mathcal{M}}\}$, and $\mathbf{D} = \mathbf{Q}_2^\top \mathbf{P}\mathbf{Q}_2$. In the following, we introduce the bivariate penalized spline over triangulation (BPST) smoothed M-estimator (M-BPST estimator) of the mean function as

$$\widehat{\mu}(\boldsymbol{z}) = \widetilde{\mathbf{B}}^\top(\boldsymbol{z})\widehat{\boldsymbol{\theta}}, \tag{2.5}$$

where $\widehat{\boldsymbol{\theta}}$ is the minimizer of (2.4).

To find the solution to (2.4), denote the derivative of $\rho(\cdot)$ by $\psi(\cdot)$. For convex ρ, such as Huber's function, the unique solution $\widehat{\boldsymbol{\theta}}$ of (2.4) can be characterized as the solution of

$$\sum_{i=1}^{n}\sum_{j=1}^{N} \psi\left\{Y_{ij} - \widetilde{\mathbf{B}}^\top(\boldsymbol{z}_j)\widehat{\boldsymbol{\theta}}\right\}\widetilde{\mathbf{B}}(\boldsymbol{z}_j) - h\mathbf{D}\widehat{\boldsymbol{\theta}} = \mathbf{0}. \tag{2.6}$$

Let $\widehat{e}_{ij}(\boldsymbol{\theta}) = Y_{ij} - \widetilde{\mathbf{B}}^\top(\boldsymbol{z}_j)\boldsymbol{\theta}$ be the residuals associated with coefficient $\boldsymbol{\theta}$, and define the

weights $w_{ij}(\boldsymbol{\theta}) = \psi\{\widehat{e}_{ij}(\boldsymbol{\theta})\}/\widehat{e}_{ij}(\boldsymbol{\theta})$. Then (2.6) can be expressed as

$$\sum_{i=1}^{n}\sum_{j=1}^{N} w_{ij}(\widehat{\boldsymbol{\theta}})\widehat{e}_{ij}(\widehat{\boldsymbol{\theta}})\widetilde{\mathbf{B}}\left(\boldsymbol{z}_j\right) - h\mathbf{D}\widehat{\boldsymbol{\theta}} = \mathbf{0}, \tag{2.7}$$

which suggests an iteratively reweighted least squares procedure (Maronna et al., 2019) to compute the solution to (2.4). Given the estimate $\widehat{\boldsymbol{\theta}}^{(\ell)}$ for the ℓ-th iteration, we compute the weights $w_{ij}^{(\ell)} = w_{ij}(\widehat{\boldsymbol{\theta}}^{(\ell)})$, and update the estimate $\widehat{\boldsymbol{\theta}}^{(\ell+1)}$ by solving (2.7). Letting $\mathbf{W}^{(\ell)} = \text{diag}(\sum_{i=1}^{n} w_{i1}^{(\ell)}, \ldots, \sum_{i=1}^{n} w_{iN}^{(\ell)})$, $S_j^{(\ell)} = \sum_{i=1}^{n} w_{ij}^{(\ell)} Y_{ij}$, and $\mathbf{S}^{(\ell)} = (S_1^{(\ell)}, \ldots, S_N^{(\ell)})^\top$, the solution $\widehat{\boldsymbol{\theta}}^{(\ell+1)}$ is of the form

$$\widehat{\boldsymbol{\theta}}^{(\ell+1)} = \left(\widetilde{\mathbf{B}}^\top \mathbf{W}^{(\ell)} \widetilde{\mathbf{B}} + h\mathbf{D}\right)^{-1} \widetilde{\mathbf{B}}^\top \mathbf{S}^{(\ell)}. \tag{2.8}$$

The procedure is repeated until a pre-specified stopping criterion is met. Denote the estimate from the iterative algorithm as $\widehat{\boldsymbol{\theta}}$. Then the estimated mean function is $\widehat{\mu}(\boldsymbol{z}) = \widetilde{\mathbf{B}}^\top(\boldsymbol{z})\widehat{\boldsymbol{\theta}}$.

3. Asymptotic Properties of the Estimator

In this section, we discuss the asymptotic consistency and normality of the proposed estimator. We start by introducing some notation. For a real-valued function g, denote by $\|g\|_2$ the standard L_2 space norm, i.e., $\|g\|_2^2 = \int_\Omega g^2(\boldsymbol{z})\, d\boldsymbol{z}$, and by $\|g\|_\infty = \sup_{\boldsymbol{z}\in\Omega} |g(\boldsymbol{z})|$ the supremum norm of g. Let $|g|_{v,\infty,\Omega} = \max_{i+j=v} \|\nabla_{z_1}^i \nabla_{z_2}^j g\|_{\infty,\Omega}$ be the maximum norm of all the v-th order derivatives of g over Ω. In addition, for any vector $\boldsymbol{v} = (v_1, \ldots, v_k) \in \mathbb{R}^k$, let $\|\boldsymbol{v}\|_2^2 = \sum_{i=1}^{k} |v_i|^2$; for a matrix $\mathbf{A}$, $\|\mathbf{A}\|_2 = \sup_{\boldsymbol{v}\neq 0} \|\mathbf{A}\boldsymbol{v}\|_2/\|\boldsymbol{v}\|_2$. Let $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ be the largest and smallest eigenvalues of matrix $\mathbf{A}$, respectively. Note that $\|\mathbf{A}\|_2 = \lambda_{\max}(\mathbf{A})$, and if the matrix $\mathbf{A}$ is non-singular, $\|\mathbf{A}^{-1}\|_2 = \lambda_{\min}^{-1}(\mathbf{A})$.

Let $\|\cdot\|_\mathcal{E}$ be the norm induced by the inner product $\langle\cdot,\cdot\rangle_\mathcal{E}$, where, for $g_1(\boldsymbol{z})$ and $g_2(\boldsymbol{z})$,

$$\langle g_1, g_2\rangle_\mathcal{E} = \int_\Omega \left\{\sum_{i+j=2} \binom{2}{i} \left(\nabla_{z_1}^i \nabla_{z_2}^j g_1(\boldsymbol{z})\right)^2\right\}^{1/2} \left\{\sum_{i+j=2} \binom{2}{i} \left(\nabla_{z_1}^i \nabla_{z_2}^j g_2(\boldsymbol{z})\right)^2\right\}^{1/2} d_{z_1} d_{z_2}.$$

Without loss of generality, we assume that the area of the domain Ω is 1. Throughout the section, we use c, C as generic constants, which may be different even in the same line.

In the following, we introduce some technical assumptions:

(A1) The triangulation is π-quasi-uniform, i.e., there exists a positive constant π such that $(\min_{\tau\in\triangle}\varpi_\tau)^{-1}|\triangle|\leq\pi$, where ϖ_τ is the radius of the largest disk that can be inscribed in $\tau\in\triangle$.

(A2) The bivariate function $\mu\in\mathcal{W}^{d+1,\infty}(\Omega)=\{g:|g|_{v,\infty,\Omega}<\infty,0\leq v\leq d+1\}$ for an integer $d\geq 1$. špace-6pt

(A3) The triangulation size satisfies that $|\triangle|\to 0$ and $|\triangle|\gg\max\{n^{-1/6},Nn^{-1/2}\}$. The degree of polynomials $d\geq 3$. In addition, $N\gg\log n$. The roughness penalty parameter satisfies that $h=o(\sqrt{n}N|\triangle|^3)$.

(A4) ρ is a convex function satisfying $\rho(0)=0$, and it is continuous, even, and non-decreasing in $[0,\infty)$. Furthermore, its derivative $\psi(\cdot)$ is continuous, non-decreasing, and uniformly bounded, i.e., $|\psi(x)|<C$ for all $x\in\mathbb{R}$.

(A5) $\mathbb{E}\psi(e_{ij})=0$.

(A6) There exists a bounded function $\delta(\boldsymbol{z})$ that satisfies $0<\inf_{\boldsymbol{z}\in\Omega}\delta(\boldsymbol{z})<\sup_{\boldsymbol{z}\in\Omega}\delta(\boldsymbol{z})<\infty$, such that $|\mathbb{E}\{\psi(e_{ij}+u)\}-\delta(\boldsymbol{z}_j)\cdot u|\leq Cu^2$, where $|u|<C$.

(A7) $\mathbb{E}\{\psi(e_{ij}+u)-\psi(e_{ij})\}^2\leq C|u|$, and $|\psi(u+v)-\psi(v)|<C$, for $|u|<C$, and $v\in\mathbb{R}$.

(A8) Define $\boldsymbol{e}_i=(e_{i1},\ldots,e_{iN})^\top$, $\boldsymbol{\psi}(\boldsymbol{e}_i)=\{\psi(e_{i1}),\ldots,\psi(e_{iN})\}^\top$, and $\mathbf{K}_i=\mathbb{E}\{\boldsymbol{\psi}(\boldsymbol{e}_i)\boldsymbol{\psi}(\boldsymbol{e}_i)^\top\}$, $1\leq i\leq n$. We assume that $\min_{1\leq i\leq n}\lambda_{\min}(\mathbf{K}_i)\geq\lambda_0>0$.

Assumption (A1) is typical for BPST-based estimation methods, which suggests the use of a more uniform triangulation with smaller shape parameters, and Assumption (A2) is typical for the true underlying functions in the nonparametric estimation literature (Lai and Wang, 2013; Yu et al., 2020; Wang et al., 2020). In our asymptotic framework, both the number of subjects (n) and the number of pixels inside the domain (N) grow large to indicate larger sample sizes and higher image resolutions. Assumption (A3) describes the requirement for the growth rate of the dimension of the spline space relative to n and N. Assumption (A4)

guarantees the existence of a unique solution to the optimization problem (2.4). Assumptions (A5) to (A7) are regularity conditions for $\psi(\cdot)$ and errors e_{ij}. Assumption (A5) ensures that the M-estimator is properly centered and unbiased under the true error distribution. Assumption (A6) imposes local linearity, enabling asymptotic normality by ensuring that the influence function dominates higher-order terms. Assumption (A7) controls the variability of ψ via bounded increments and Lipschitz continuity, which is critical for uniform convergence and robustness. For the Huber loss in (2.2), the derivative $\psi_k(x) = \max\{-k, \min\{x, k\}\}$ satisfies all these conditions, as discussed in Kalogridis and Van Aelst (2023a). Assumption (A8) is needed for the asymptotic normality. Assumptions (A5) to (A8) are typical in the literature on M-type spline regression (Wei and He, 2006; Tang and Cheng, 2008; Lima et al., 2019a).

Theorem 1. *(Asymptotic consistency) Under Assumptions (A1)–(A7), if* $N^{1/2}|\triangle| \to \infty$ *as* $N \to \infty$*, then the M-BPST estimator* $\widehat{\mu}(\cdot)$ *in (2.5) is consistent and satisfies* $\|\widehat{\mu} - \mu\|_2 = O_p\left(|\triangle|^{-1}n^{-1/2} + |\triangle|^{d+1}\right)$.

Remark 1. *By selecting* $|\triangle| \asymp n^{-1/\{2(d+2)\}}$*, an optimal convergence rate* $O_P(n^{-(d+1)/\{2(d+2)\}})$ *(Stone, 1985) can be achieved.*

To state the asymptotic normality result, we first define

$$\mathbf{L}_{n,h} = \frac{n}{N}\sum_{j=1}^{N}\delta(\boldsymbol{z}_j)\widetilde{\mathbf{B}}(\boldsymbol{z}_j)\widetilde{\mathbf{B}}^\top(\boldsymbol{z}_j) + \frac{h}{N}\mathbf{Q}_2^\top\left[\langle B_m, B_{m'}\rangle_{\mathcal{E}}\right]_{m,m'\in\mathcal{M}}\mathbf{Q}_2, \tag{3.1}$$

where $\delta(\cdot)$ is defined in the assumption (A6).

Theorem 2. *(Asymptotic normality) Under Assumptions (A1)–(A8), if* $N^{1/2}|\triangle| \to \infty$ *as* $N \to \infty$ *and* $|\triangle| \ll n^{-1/\{2(d+2)\}}$*,* $\{\boldsymbol{\Sigma}_n(\boldsymbol{z})\}^{-1/2}\{\widehat{\mu}(\boldsymbol{z}) - \mu(\boldsymbol{z})\} \xrightarrow{d} N(0,1)$*,* $\boldsymbol{z} \in \Omega$*, where*

$$\boldsymbol{\Sigma}_n(\boldsymbol{z}) = \widetilde{\mathbf{B}}^\top(\boldsymbol{z})\mathbf{L}_{n,h}^{-1}\left(\sum_{i=1}^{n}N^{-2}\widetilde{\mathbf{B}}^\top\mathbf{K}_i\widetilde{\mathbf{B}}\right)\mathbf{L}_{n,h}^{-1}\widetilde{\mathbf{B}}(\boldsymbol{z}), \tag{3.2}$$

and $\mathbf{K}_i$ *is defined in the Assumption (A8).*

Remark 2. *When* $|\triangle| \ll n^{-1/\{2(d+2)\}}$*, the bias of* $\widehat{\mu}(\cdot)$ *is asymptotically negligible. Hence, the result of Theorem 2 can be used to construct point-wise confidence intervals.*

Remark 3. *In practice, to estimate* $\mathbf{\Sigma}_n(\boldsymbol{z})$*, we first obtain the error estimates* $\widehat{e}_{ij} = Y_{ij} - \widehat{\mu}(\boldsymbol{z}_j)$*. Then,* $\mathbf{K}_i$ *can be estimated as* $\widehat{\mathbf{K}}_i = \boldsymbol{\psi}(\widehat{\boldsymbol{e}}_i)\boldsymbol{\psi}(\widehat{\boldsymbol{e}}_i)^\top$*, where* $\boldsymbol{\psi}(\widehat{\boldsymbol{e}}_i) = \{\psi(\widehat{e}_{i1}), \ldots, \psi(\widehat{e}_{iN})\}^\top$*. Using the Huber loss function,* $\widehat{\delta}(\boldsymbol{z}_j) = n^{-1}\sum_{i=1}^{n} I(|\hat{e}_{ij}| \leq k)$ *where* k *is the constant in* (2.2)*. Finally,* $\mathbf{\Sigma}_n(\boldsymbol{z})$ *can be estimated by plugging* $\widehat{\delta}(\cdot)$ *and* $\widehat{\mathbf{K}}_i$ *into* (3.1) *and* (3.2)*, respectively.*

4. Robust Simultaneous Confidence Corridors

Constructing valid simultaneous confidence corridors (SCCs) for functional or imaging data requires addressing two key challenges: the spatial dependencies that arise across the domain and the potential for heavy-tailed errors. Although Theorem 2 provides the asymptotic normality needed for robust pointwise confidence intervals, it does not immediately extend to SCCs. We address this gap by proposing a weighted bootstrap procedure that utilizes the pointwise variance estimates from Remark 3, thereby facilitating robust SCC construction across the entire domain. While weighted bootstrap is widely used for simultaneous inference in M-type regression settings (Belloni et al., 2019; Chen and Zhou, 2020; Liu et al., 2023), its application to robust SCCs for functional and imaging data has remained largely unexplored.

Common bootstrap strategies in imaging contexts, such as the wild bootstrap (Yu et al., 2021; Gu et al., 2025), are well-suited to heteroscedastic Gaussian errors yet can fail under heavy-tailed distributions. Its reliance on random weights (e.g., sign flips via Rademacher variables ± 1) tends to amplify the influence of large residuals, leading to distorted bootstrap distributions and inflated variance estimates, resulting in suboptimal coverage properties in the presence of outliers. Additionally, the wild bootstrap implicitly assumes symmetric error distributions when using symmetric weights, an assumption often violated by skewed heavy-tailed data. This mismatch between bootstrap perturbations and true error structures further biases inference. In contrast, the weighted bootstrap paired with Huber regression offers a robust alternative. By assigning i.i.d. random weights with unit mean/variance to observations, this approach smoothly modulates the influence of outliers rather than amplifying them through discrete perturbations. Huber regression's inherent robustness, achieved by

downweighting large residuals with appropriately chosen tuning constants, complements the weighted bootstrap's flexibility. In addition, because the weighted bootstrap preserves spatial or functional dependencies inherent in structured data (e.g., neuroimaging voxels), it avoids artificial discontinuities introduced by resampling.

After obtaining the mean function and pointwise variance estimates $\widehat{\mu}(\cdot)$ and $\widehat{\mathbf{\Sigma}}_n(\cdot)$, we draw i.i.d. random weights $\iota_1, \iota_2, \ldots, \iota_n$, from $\text{Exp}(1)$, an exponential distribution with scale parameter 1, ensuring nonnegative weights with unit mean and variance. These properties contribute to numerical stability and maintain the convexity of the Huber loss minimization. Although $\text{Exp}(1)$ is a standard choice, one can use any distribution satisfying unit mean and variance, such as $N(0,1)+1$ or $2\times \text{Ber}(0.5)$, as discussed in Chen and Zhou (2020). However, the use of nonnegative weights has the added advantage of guaranteeing convexity in the weighted objective function. At each bootstrap iteration, we solve

$$\widehat{\boldsymbol{\theta}}^b = \underset{\boldsymbol{\theta}\in\mathbb{R}^{|\widetilde{\mathcal{M}}|}}{\arg\min} \sum_{i=1}^{n} \iota_i \sum_{j=1}^{N} \rho\left\{Y_{ij} - \widetilde{\mathbf{B}}^\top(\boldsymbol{z}_j)\,\boldsymbol{\theta}\right\} + \frac{h}{2}\boldsymbol{\theta}^\top \mathbf{D}\boldsymbol{\theta}, \tag{4.1}$$

and define $\widehat{\mu}^b(\boldsymbol{z}) = \widetilde{\mathbf{B}}^\top(\boldsymbol{z})\widehat{\boldsymbol{\theta}}^b$ as the bootstrap estimate of the mean function. After we obtain B bootstrap estimates, we compute the critical value $\widehat{c}_n(\alpha)$ which is defined as

$$\hat{c}_n(\alpha) = \inf\left\{c \geq 0 : \frac{1}{B}\sum_{b=1}^{B} \mathbb{I}\left(\max_{\boldsymbol{z}\in\Omega} \left|T^b(\boldsymbol{z})\right| \leq c\right) \geq 1-\alpha\right\},$$

where $T^b(\boldsymbol{z}) = \widehat{\mathbf{\Sigma}}_n(\boldsymbol{z})^{1/2}\{\widehat{\mu}^b(\boldsymbol{z}) - \widehat{\mu}(\boldsymbol{z})\}$, and construct the robust SCC as $\widehat{\mu}(\boldsymbol{z}) \pm \widehat{c}_n(\alpha)\sqrt{\widehat{\mathbf{\Sigma}}_n(\boldsymbol{z})}$, $\forall \boldsymbol{z} \in \Omega$. The detailed procedure is summarized in Algorithm 1.

5. Implementation

The triangulation and associated bivariate spline smoothing parameters, specifically the degree d and the smoothness parameter r, are crucial for estimation accuracy. When the underlying mean image exhibits smooth characteristics without sharp discontinuities, selecting $r = 1$ and $d \geq 4$ is recommended. In particular, adopting $d \geq 5$ enables the proposed spline method to achieve an asymptotic full estimation power (Lai and Wang, 2013).

Algorithm 1: Robust SCC for the mean function via weighted bootstrap

Input : Data $\{Y_{ij}\}_{i,j=1}^{n,N}$, number of bootstrap replications B, nominal level $\alpha \in (0,1)$.

Solve the penalized M-estimation in (2.4) to obtain $\{\widehat{\mu}(\boldsymbol{z})\}$ and $\{\widehat{\boldsymbol{\Sigma}}_n(\boldsymbol{z})\}$, $\boldsymbol{z} \in \Omega$;

for $b = 1, \ldots, B$ **do**

  Generate i.i.d. random weights $\iota_{i,b}$ from exponential distribution with parameter 1;

  Solve weighted penalized M-estimation in (4.1) to obtain $\widehat{\mu}^b(\boldsymbol{z})$;

  Calculate the absolute maximal deviation $\max_{\boldsymbol{z}\in\Omega} |\{\widehat{\boldsymbol{\Sigma}}_n(\boldsymbol{z})\}^{-1/2}\{\widehat{\mu}^b(\boldsymbol{z}) - \widehat{\mu}(\boldsymbol{z})\}|$.

end

Determine $\widehat{c}_n(\alpha)$ as the $(1-\alpha)$ sample quantile of the above absolute maximal deviations;

Output : Robust SCC is $\widehat{\mu}(\boldsymbol{z}) \pm \widehat{c}_n(\alpha)\sqrt{\widehat{\boldsymbol{\Sigma}}_n(\boldsymbol{z})}$, $\forall \boldsymbol{z} \in \Omega$.

Several factors influence the triangulation fineness required to estimate the mean function accurately. Primarily, the geometric complexity of the target object's domain within the image significantly influences triangulation requirements. Such domains, when characterized by high curvature or intricate morphological structures, inherently require more refined triangulation strategies. As stated in Theorem 1, the functional characteristics of the underlying signal are important in triangulation selection. Functions exhibiting rapid spatial variations or localized features require more granular triangulation to capture intricate morphological nuances accurately. Furthermore, the theorem indicates that both the number of images n and pixels N in the domain substantively affect the appropriate triangulation refinement. Although increased n and N permit finer triangulation for more precise target function estimation, numerical stability and reliable estimation of local spline coefficients requires ensuring that each triangle contains a minimum of $\lfloor \binom{d+3}{3}/2 \rfloor$ observations, dependent on the degree of spline basis d. Our numerical studies indicate that, beyond a certain refinement threshold, further triangulation refinement provides negligible improvements in estimation accuracy.

We select the penalty parameter h in (2.3) by minimizing the following weighted version of the generalized cross-validation (GCV) criterion (Kalogridis and Van Aelst, 2023a):

$$\mathrm{GCV}(h) = N^{-1}\left(1 - \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{N} H_{ij}\right)^{-2} \sum_{i=1}^{n}\sum_{j=1}^{N} w_{ij}(\widehat{\boldsymbol{\theta}}_h) e_{ij}^2(\widehat{\boldsymbol{\theta}}_h),$$

where $\widehat{\boldsymbol{\theta}}_h$ is the M-BPST estimate computed corresponding to the roughness penalty h. At

convergence of the iterative algorithm, H_{ij} measures the influence of the ij-th observation, which is obtained from the diagonal of the weighted hat matrix.

As described in Section 2.2, the mean function estimates are computed via iteratively reweighted least squares procedures. Due to the large scale of the optimization problem, which is driven by factors such as sample size, image resolution, repeated least squares computations, and roughness penalty selection via generalized cross-validation, an efficient algorithm and implementation are necessary.

To address these computational challenges, we implement the functions for robust mean estimation and smoothness parameter selection, which are the primary bottlenecks of our procedure, using the Armadillo C++ library (Sanderson and Curtin, 2016) to improve computational efficiency. This implementation is seamlessly integrated with R using the Rcpp wrapper (Eddelbuettel and Sanderson, 2014). Furthermore, equation (2.8) is calculated using the conjugate gradient algorithm (Shewchuk, 1994) whenever the weight matrix or h is updated, utilizing the proximity of solutions.

The Huber constant, k, plays a critical role in balancing robustness and efficiency when addressing noise and contamination in data. A proper choice of k should be calibrated proportionally to the scale of the noise e_{ij} and the anticipated contamination severity in the dataset. A smaller k increases robustness by down-weighting more observations with large residuals, effectively reducing the influence of outliers. However, this comes at the cost of efficiency, as non-outlying observations may be unnecessarily down-weighted. Conversely, a larger k improves efficiency but may compromise robustness, making the estimator more vulnerable to the influence of outliers. In the simulation studies presented in Section 6 and Section S1 of the Supplementary Material, all examined values of k demonstrate sufficient robustness for mean function estimation, while smaller values of k result in a slight loss of efficiency compared to the larger values. However, the impact of k becomes particularly pronounced in the proposed RSCC procedure. Specifically, smaller k values yield wider

corridors, providing better empirical coverage but resulting in more conservative inference, whereas larger k produce narrower corridors, improving statistical efficiency at the potential expense of robustness. This trade-off demonstrates the importance of choosing k based on the dataset's characteristics and the analysis's goals.

6. Simulation Studies

In this section, we conduct simulation studies to evaluate the performance of the proposed methods. The corresponding R code used in these studies is available on `GitHub` (`https://github.com/ylong-chn/Robust-ImageSCC`). The simulated imaging data are generated from the following model:

$$Y_{ij} = \mu(\boldsymbol{z}_j) + \sum_{\kappa=1}^{2} \lambda_\kappa^{1/2} \xi_{i\kappa} \psi_\kappa(\boldsymbol{z}_j) + \varepsilon_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, N, \tag{6.1}$$

where $\boldsymbol{z}_j = (z_{1j}, z_{2j}) \in \Omega \subseteq [0,1]^2$. We consider two different spatial domains: a human brain scan domain (Ω_1) and a horseshoe domain (Ω_2) with the following true mean functions.

- Quadratic function on Ω_1: $\mu(\boldsymbol{z}) = 20\{(z_1 - 0.5)^2 + (z_2 - 0.5)^2\}$;
- Exponential function on Ω_1: $\mu(\boldsymbol{z}) = 5\exp[-15\{(z_1 - 0.5)^2 + (z_2 - 0.5)^2\}] + 0.5$;
- Cubic function on Ω_1: $\mu(\boldsymbol{z}) = 3.2(-z_1^3 + z_2^3) + 2.4$;
- Sine function on Ω_1: $\mu(\boldsymbol{z}) = 3[\sin\{5\pi z_1 + 0.22\} - \sin\{5\pi(z_2 - 0.18)\}] + 4.8$;
- Reference function $\mu(\boldsymbol{z})$ on Ω_2: the test function in the `mgcv` R package (Wood, 2017).

For within-image dependence, we generate $\xi_{i\kappa} \sim N(0,1)$ independently, $\kappa = 1, 2$. The eigenstructure is defined by the eigenvalues $\lambda_1 = 0.5$ and $\lambda_2 = 0.2$, with the corresponding orthonormal eigenfunctions: $\psi_1(\boldsymbol{z}) = 0.988\sin(\pi z_1) + 0.5$ and $\psi_2(\boldsymbol{z}) = 2.157\cos(\pi z_2) - 0.039$. The constants are chosen so that the eigenfunctions are orthonormal. The random noise is specified as $\varepsilon_{ij} \sim N(0, 0.2^2)$. We consider sample sizes $n = 100,\ 200$ and 400. Each simulated image is generated at a resolution of 40×40, and the pixels are indexed by $\{1, 2, \ldots, N\}$, whereas of the 1,600 pixels, $N = 921$ are inside the domain Ω_1 and $N = 1166$ are within the domain Ω_2.

Following Wang and Cao (2022), we consider four types of contamination scenarios: two involving surface outliers and two involving heavy-tailed error distributions. All contamination types are implemented through modifications to the error term ε_{ij}. For the surface outlier cases, we assume that a proportion of the images are contaminated. Specifically, we contaminate a subset $\mathcal{R}^o$ of the original sample set $\{1, 2, \ldots, n\}$, with contamination rates of 0% (baseline), 5%, 10%, and 15%. In these cases, we replace observations Y_{ij^*} with contaminated values $Y^o_{ij^*}$ for $i \in \mathcal{R}^o$, where j^* denotes pixels in the contaminated region, i.e. $j^* \in \mathcal{O} \subset \{1, 2, \ldots, N\}$. The contaminated values $Y^o_{ij^*}$ are generated using the same mean function and eigenfunctions as the original samples but with modified noise term $\varepsilon^o_{ij^*}$. For the heavy-tailed error cases, the noise term ε_{ij} follows a mixture of $N(0, 0.2^2)$ and a heavy-tailed distribution with mixing weights of 5%, 10%, and 15% assigned to the heavy-tailed component. The specific parametrization for each contamination scenario is detailed below.

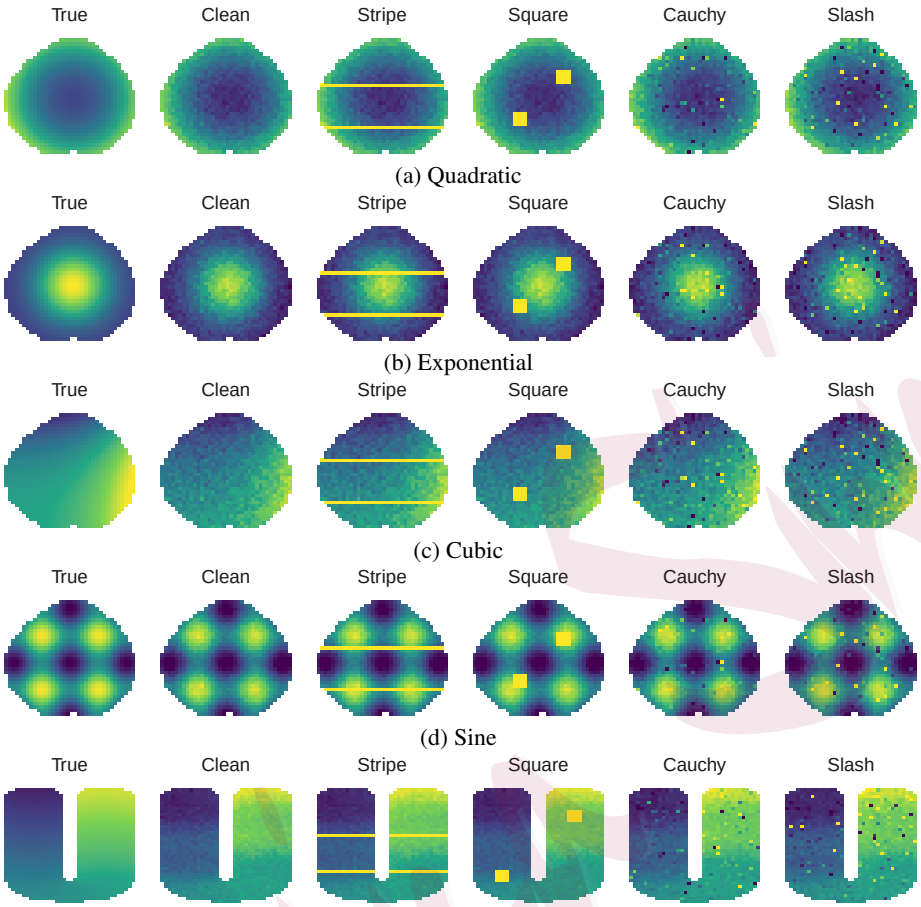
Case 1: *Stripe outliers.* Contamination occurs along line segments with $\varepsilon^o_{ij^*} \sim U(10, 20)$.

Case 2: *Square outliers.* Contamination occurs in squared regions with $\varepsilon^o_{ij^*} \sim U(10, 20)$.

Case 3: *Mixture Normal–Cauchy.* Errors ε_{ij} follow a heavy-tailed distribution, which comprises a mixture of $N(0, 1)$ and a Cauchy distribution with location 0 and scale 0.5.

Case 4: *Mixture Normal–Slash.* Errors ε_{ij} follow a heavy-tailed distribution, which comprises a mixture of $N(0, 1)$ and a Slash distribution with location 0 and scale 0.5.

Figure 1 illustrates examples of images generated using the aforementioned function types, with various types of contamination applied. There are six columns in Figure 1: The first two columns display the true mean function and a sample image, respectively. The following four columns show the four types of contamination for each mean function. The color scale represents function values, with darker hues indicating lower values and lighter hues indicating higher values. Note that outlier values have been compressed within the visualization's color range to maintain interpretability across all scenarios. The mixing weight for the heavy-tailed distributions is set to 15% for the Cauchy and Slash examples.



(a) Quadratic

(b) Exponential

(c) Cubic

(d) Sine

(e) Reference function in Wood (2017)

Figure 1: True mean functions and observed images on a scan of brain-shape domain or horseshoe domain with various types of contamination effects. The first three rows present a different functional form on a scan of the brain-shape domain: (a) quadratic, (b) exponential, (c) cubic, and (d) sine. The last row is based on a mean function with specification in the `mgcv` R package (Wood, 2017). Columns display (from left to right): true mean function $\mu(\boldsymbol{z})$, uncontaminated sample image, and four contamination scenarios (stripe outliers, square outliers, 15% Normal-Cauchy mixture, 15% Normal-Slash mixture).

6.1. Simulation Study on Robust Mean Function Estimation

In this section, we compare the performance of the following estimators: the non-robust BPST-smoothed mean function estimator (Wang et al., 2020); the robust deep neural network (RDNN) estimator (Wang and Cao, 2022); the proposed M-BPST estimator; tensor-product spline with B-splines (TPB); the thin plate smoothing estimator (TPS, Wood, 2003) for the brain-shaped domain; and the soap film smoothing estimator (SFS, Wood et al., 2008) for the horseshoe domain. The TPB, TPS, and SFS are implemented in the `mgcv` package. To accommodate irregular domains, we adapt the RDNN implementation to accept domain-

specific coordinates as input variables. For both the M-BPST and RDNN, we employ the Huber loss with a tuning constant of 0.1, a conservative choice based on the noise level and contamination settings used in the simulation. While this ensures robustness, it comes at the expense of some efficiency. Further simulations exploring the sensitivity of the M-BPST to different values of k are provided in Section S1.1 of the Supplementary Material. These results demonstrate that all the tested k values provide sufficient robustness, with only minor differences in efficiency observed.

The default settings are used in the RDNN algorithm for parameters such as batch size, number of hidden layers, number of neurons, dropout rate, and number of epochs. For TPS and TPB, the number of basis functions is chosen to match the degrees of freedom of the BPST methods. For SFS, we define the boundary as the horseshoe-domain contour and use all points inside the boundary as interior knots. Regarding BPST and M-BPST estimators, the triangulations used for the brain and horseshoe domains are illustrated in Figures 2 and 3, respectively. The brain domain triangulation comprises 80 triangles with 54 vertices, while the horseshoe domain triangulation consists of 112 triangles and 88 vertices. The degree parameter d for spline basis functions is set to 5. To assess the sensitivity of the method, an alternative triangulation and degrees $d = 4, 6$ are also explored. Our numerical investigations indicate that the proposed method demonstrates considerable robustness to variations in triangulation and degree selection. A detailed sensitivity analysis is available in Section S1.1 of the Supplementary Material.

The comparative performance of the estimators under various contamination scenarios is shown in Figures 2 and 3. Figure 2 demonstrates their behavior using simulated data generated from a sine mean function on the domain Ω_1, while Figure 3 presents results based on the test function implemented in the `mgcv` R package. Due to space limit, further illustrations using data generated from an exponential, cubic and quadratic mean function on Ω_1 are included in Supplementary Section S1.2. In Figure 3, the TPB exhibits pronounced

boundary "leakage" on the highly irregular horseshoe domain, whereas the other methods remain unaffected. Moreover, the non-robust estimators (BPST, SFS and TPB) degrade under all four contamination types, while M-BPST and RDNN maintain robust performance in the presence of extreme values.

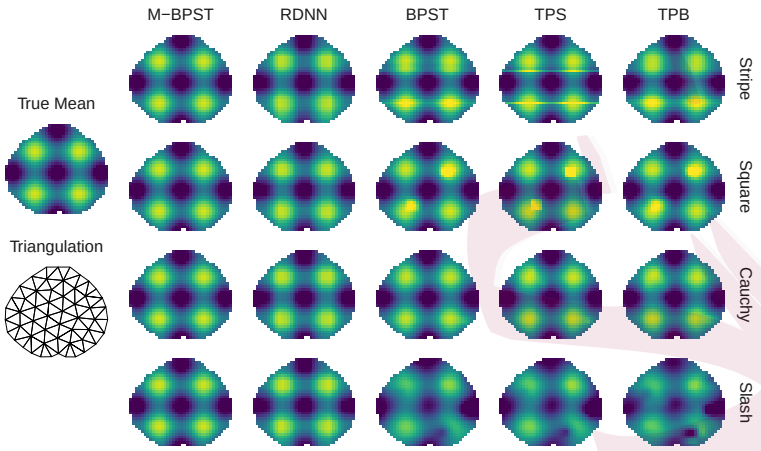


Figure 2: Simulation run demonstrating the performance of mean function estimation methods on a Sine mean function. The leftmost column displays the true mean function alongside the triangulation setting used for the BPST and M-BPST estimators. The subsequent five columns present different estimators based on a typical replication with $n = 200$ under four contamination scenarios: stripe and square contamination at a 15% contamination rate, 15% Normal-Cauchy mixture, and 15% Normal-Slash mixture.

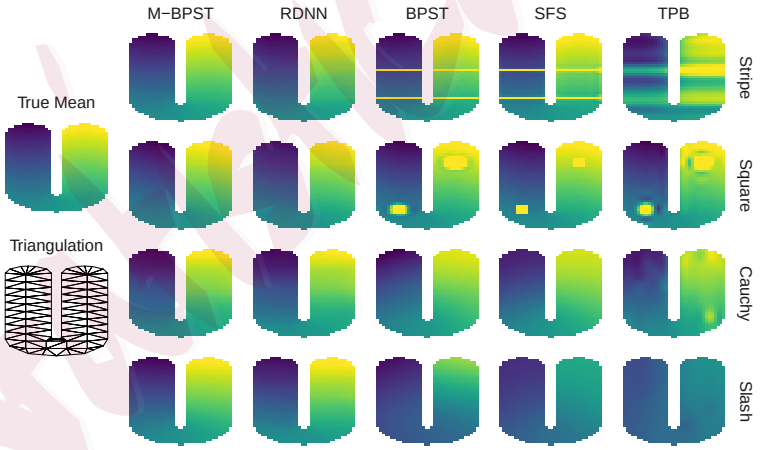


Figure 3: Simulation run demonstrating the performance of mean function estimation methods on a **horseshoe** mean function. The leftmost column displays the true mean function alongside the triangulation setting used for the BPST and M-BPST estimators. The subsequent five columns present different estimators based on a typical replication with $n = 200$ under four contamination scenarios: stripe and square contamination at a 15% contamination rate, 15% Normal-Cauchy mixture, and 15% Normal-Slash mixture.

Tables 1 and 2 present results from 100 replications evaluating the mean function estimators for images generated with a sine mean function on Ω_1 and the test function from `mgcv` on Ω_2, respectively. Additional results for images generated with exponential, cubic

and quadratic mean functions on Ω_1 are provided in Supplementary Section S1.2. The tables report the average and standard deviation of the mean integrated squared errors (MISE) under both uncontaminated and contaminated scenarios. In the uncontaminated (clean) case, the BPST and M-BPST estimators exhibit slightly faster convergence rates and achieve lower average MISE compared to the RDNN estimator across all settings. Due to the conservative choice of the Huber constant, the M-BPST estimator has a marginally higher average MISE than the BPST estimator. As shown in Supplementary Section S1.1, when contamination is not significantly detrimental, using a higher value of k can maintain sufficient robustness while achieving a smaller MISE compared to more conservative values of k. The TPB estimator yields results similar to those of the BPST estimator. Likewise, TPS on Ω_1 and SFS on Ω_2 yield comparable performance; however, the BPST offers a slight advantage in handling irregular domains, which is particularly evident in the horseshoe domain.

As stripe or square-type contamination is introduced, the average error for the non-robust estimators (BPST, TPS, SFS, and TPB) increases with the contamination rate, whereas both robust estimators show nearly constant error levels, indicating a clear advantage. Note that under contamination, BPST's MISE can even exceed that of the `mgcv` smoothers because the impact of outliers can offset BPST's inherent strengths. Among the robust methods, the M-BPST estimator has a small but consistent edge over the RDNN estimator. In contaminated settings, increasing the sample size does not reduce the average MISE for the non-robust method, unlike for the two robust estimators. For heavy-tailed error distributions, the benefit of robust methods is more evident, as the three non-robust estimators are highly affected by extreme-valued pixels in the images. Both Tables 1 and 2 show that the non-robust estimators suffer from severe inconsistency due to these large errors, whereas both robust methods maintain low average MISE and benefit from larger sample sizes. Across all scenarios, the M-BPST estimator yields a lower average MISE than the RDNN estimator, highlighting the effectiveness of BPST-smoothed methods on irregular domains.

Another notable observation in Tables 1 and 2 is that the performance of RDNN can sometimes stagnate when the sample size n increases to 400, as the MISE does not necessarily improve. This may be attributed to the simplicity of the architecture used in the simulation. Although we also explored more complex architectures, including the deeper designs recommended in Wang and Cao (2022), these experiments did not yield improvements over the default configuration. This highlights the inherent challenges in hyperparameters tuning for the RDNN estimator, which can impact its performance in practical applications.

Table 1: Mean (and SD) of integrated squared errors ($\times 100$) of different estimators across various scenarios for the **sine** mean function. Bold values indicate the best-performing estimator in each scenario.

Cont.	$n = 100$					$n = 200$					$n = 400$				
	M-BPST	RDNN	BPST	TPS	TPB	M-BPST	RDNN	BPST	TPS	TPB	M-BPST	RDNN	BPST	TPS	TPB
Stripe															
0.05	**2.05** (**1.85**)	15.0 (3.44)	2.88 (1.63)	5.25 (1.64)	3.13 (1.63)	**0.93** (**0.76**)	12.2 (2.57)	2.27 (1.10)	4.65 (1.10)	2.52 (1.10)	**0.42** (**0.35**)	10.9 (2.73)	1.93 (0.64)	4.32 (0.64)	2.18 (0.64)
0.10	**2.06** (**1.87**)	15.4 (3.41)	7.86 (2.34)	17.4 (2.36)	8.03 (2.34)	**0.97** (**0.77**)	12.4 (2.88)	7.25 (1.77)	16.9 (1.77)	7.43 (1.77)	**0.47** (**0.38**)	11.1 (2.57)	6.90 (1.14)	16.5 (1.15)	7.08 (1.14)
0.15	**2.10** (**1.91**)	15.5 (5.05)	16.2 (3.21)	37.8 (3.23)	15.8 (3.21)	**1.05** (**0.79**)	12.4 (2.85)	15.6 (2.52)	37.3 (2.53)	15.3 (2.51)	**0.57** (**0.42**)	10.7 (2.10)	15.3 (1.66)	37.0 (1.69)	14.9 (1.66)
Square															
0.05	**2.06** (**1.84**)	14.8 (3.11)	2.91 (1.43)	3.09 (1.43)	2.90 (1.43)	**0.93** (**0.75**)	11.8 (2.59)	2.29 (0.83)	2.48 (0.83)	2.28 (0.83)	**0.42** (**0.34**)	11.2 (3.05)	1.95 (0.42)	2.15 (0.42)	1.95 (0.42)
0.10	**2.07** (**1.85**)	14.7 (3.35)	6.99 (1.62)	8.66 (1.64)	7.29 (1.63)	**0.96** (**0.76**)	12.3 (2.50)	6.37 (1.08)	8.06 (1.09)	6.67 (1.08)	**0.46** (**0.35**)	10.7 (2.54)	6.02 (0.62)	7.72 (0.62)	6.33 (0.62)
0.15	**2.11** (**1.86**)	15.0 (3.91)	14.2 (1.92)	18.0 (1.94)	14.5 (1.93)	**1.03** (**0.76**)	12.4 (3.52)	13.6 (1.40)	17.4 (1.40)	13.8 (1.40)	**0.54** (**0.37**)	10.6 (2.29)	13.2 (0.86)	17.0 (0.87)	13.5 (0.86)
Cauchy															
0.05	**2.20** (**2.16**)	16.3 (4.25)	8.2e3 (7.7e4)	1.6e3 (1.4e4)	4.0e3 (3.8e4)	**0.92** (**0.74**)	12.2 (2.37)	286 (1.3e3)	128 (556)	269 (1.5e3)	**0.41** (**0.33**)	10.6 (2.93)	1.6e3 (1.5e4)	1.5e3 (1.5e4)	947 (8.8e3)
0.10	**2.20** (**2.16**)	15.1 (3.44)	3.3e4 (3.1e5)	6.2e3 (5.7e4)	1.6e4 (1.5e5)	**0.92** (**0.74**)	12.7 (2.96)	1.1e3 (4.9e3)	448 (2.0e3)	1.0e3 (5.8e3)	**0.43** (**0.38**)	10.9 (2.83)	6.2e3 (5.9e4)	6.0e3 (5.8e4)	3.7e3 (3.5e4)
0.15	**2.20** (**2.16**)	15.0 (3.61)	7.4e4 (7.0e5)	1.4e4 (1.3e5)	3.5e4 (3.4e5)	**0.92** (**0.73**)	11.6 (2.06)	2.4e3 (1.1e4)	952 (4.5e3)	2.1e3 (1.2e4)	**0.43** (**0.38**)	11.0 (2.35)	1.4e4 (1.3e5)	1.3e4 (1.3e5)	8.2e3 (7.9e4)
Slash															
0.05	**2.05** (**1.81**)	14.9 (3.83)	157 (771)	87.0 (313)	140 (603)	**0.96** (**0.86**)	12.3 (2.37)	1.8e3 (1.1e4)	967 (9.0e3)	529 (4.5e3)	**0.41** (**0.33**)	11.3 (2.52)	7.7e3 (5.5e4)	3.2e3 (2.6e4)	2.3e3 (1.6e4)
0.10	**2.05** (**1.78**)	15.9 (4.14)	541 (3.0e3)	223 (824)	439 (2.2e3)	**0.97** (**0.86**)	11.8 (2.28)	7.1e3 (4.5e4)	3.8e3 (3.6e4)	2.0e3 (1.8e4)	**0.44** (**0.38**)	11.3 (2.70)	3.1e4 (2.2e5)	1.3e4 (1.0e5)	9.0e3 (6.4e4)
0.15	**2.04** (**1.76**)	15.6 (3.19)	1.1e3 (6.7e3)	422 (1.7e3)	898 (5.0e3)	**0.97** (**0.86**)	12.4 (2.57)	1.6e4 (1.0e5)	8.3e3 (8.0e4)	4.4e3 (4.0e4)	**0.46** (**0.43**)	10.8 (2.32)	6.9e4 (5.0e5)	2.9e4 (2.4e5)	2.0e4 (1.4e5)
Clean	2.07 (1.83)	15.7 (4.47)	**1.24** (**1.39**)	1.25 (1.39)	1.43 (1.39)	0.92 (0.75)	12.0 (2.82)	**0.62** (**0.71**)	0.63 (0.71)	0.82 (0.71)	0.41 (0.33)	10.9 (2.56)	**0.29** (**0.31**)	0.29 (0.31)	0.48 (0.31)

Regarding scalability, our numerical studies indicate that RDNN runs considerably slower than both BPST and M-BPST, and that its runtime grows quickly with sample size. In contrast, although M-BPST requires multiple weighted least-squares fits, it maintains a modest runtime compared to BPST due to efficient C++ implementation and the use of a conjugate gradient algorithm. The primary computational bottleneck for M-BPST is solving the weighted least-squares system at each iteration, and the dimension of this system is determined solely by the number of basis functions $|\widetilde{\mathcal{M}}|$ (which depends on the triangulation design and the polynomial degree d) and is independent of both the number of subjects n and the number

Table 2: Average (and SD) of integrated squared errors ($\times 100$) of different estimators for the **horseshoe** mean function. Bold values indicate the best-performing estimator in each scenario.

Cont.	$n = 100$					$n = 200$					$n = 400$				
	M-BPST	RDNN	BPST	SFS	TPB	M-BPST	RDNN	BPST	SFS	TPB	M-BPST	RDNN	BPST	SFS	TPB
Stripe															
0.05	**1.09** (**1.05**)	1.61 (1.04)	4.03 (0.91)	3.81 (0.90)	2.27 (0.88)	**0.53** (**0.37**)	1.13 (0.60)	3.55 (0.43)	3.35 (0.42)	1.82 (0.40)	**0.27** (**0.16**)	0.84 (0.37)	3.35 (0.21)	3.16 (0.21)	1.64 (0.20)
0.10	**1.10** (**1.06**)	1.60 (1.12)	13.6 (1.05)	12.81 (1.03)	4.14 (0.98)	**0.54** (**0.37**)	1.16 (0.55)	13.1 (0.57)	12.4 (0.56)	3.70 (0.54)	**0.27** (**0.17**)	0.89 (0.43)	12.9 (0.34)	12.2 (0.34)	3.51 (0.32)
0.15	**1.11** (**1.07**)	1.67 (1.23)	29.6 (1.23)	27.83 (1.20)	7.44 (1.13)	**0.55** (**0.39**)	1.16 (0.46)	29.1 (0.75)	27.3 (0.76)	7.01 (0.71)	**0.30** (**0.18**)	0.86 (0.39)	29.0 (0.49)	27.2 (0.48)	6.83 (0.44)
Square															
0.05	**1.10** (**1.05**)	1.62 (1.06)	1.97 (0.94)	2.23 (0.93)	2.71 (0.92)	**0.54** (**0.37**)	1.14 (0.60)	1.51 (0.49)	1.83 (0.45)	2.25 (0.47)	**0.27** (**0.16**)	0.95 (0.48)	1.35 (0.26)	1.72 (0.21)	2.07 (0.23)
0.10	**1.10** (**1.06**)	1.57 (1.11)	5.55 (1.17)	6.31 (1.11)	6.11 (1.11)	**0.54** (**0.37**)	1.14 (0.60)	5.04 (0.74)	5.84 (0.67)	5.64 (0.67)	**0.28** (**0.17**)	0.93 (0.42)	4.92 (0.57)	5.67 (0.37)	5.45 (0.38)
0.15	**1.11** (**1.06**)	1.65 (1.13)	11.5 (1.54)	13.1 (1.36)	11.8 (1.35)	**0.55** (**0.38**)	1.24 (0.55)	10.9 (1.25)	12.6 (0.90)	11.3 (0.90)	**0.29** (**0.18**)	1.01 (0.47)	10.7 (0.96)	12.4 (0.55)	11.1 (0.54)
Cauchy															
0.05	**1.15** (**1.02**)	1.58 (1.11)	1.3e3 (7.9e3)	192 (1.7e3)	308 (2.1e3)	**0.54** (**0.36**)	1.22 (0.68)	2.4e4 (2.3e5)	54.4 (339)	3.6e3 (3.5e4)	**0.26** (**0.16**)	0.92 (0.43)	140 (1.0e3)	17.6 (61.0)	194 (1.2e3)
0.10	**1.15** (**1.00**)	1.67 (1.22)	5.2e3 (3.1e4)	722 (6.7e3)	1.2e3 (8.1e3)	**0.55** (**0.36**)	1.18 (0.55)	9.4e4 (9.3e5)	217 (1.4e3)	1.4e4 (1.4e5)	**0.26** (**0.16**)	0.90 (0.43)	550 (4.1e3)	59.2 (212)	745 (4.7e3)
0.15	**1.15** (**0.99**)	1.63 (1.11)	1.2e4 (7.1e4)	1.6e3 (1.5e4)	2.6e3 (1.8e4)	**0.55** (**0.36**)	1.14 (0.54)	2.1e5 (2.1e6)	1.3e3 (1.1e4)	3.2e4 (3.2e5)	**0.26** (**0.16**)	0.90 (0.42)	1.2e3 (9.2e3)	117 (435)	1.7e3 (1.0e4)
Slash															
0.05	**1.13** (**1.00**)	1.76 (1.26)	472 (2.7e3)	14.8 (37.3)	55.1 (207)	**0.55** (**0.37**)	1.10 (0.59)	4.2e5 (4.2e6)	6.1e3 (5.4e4)	6.7e4 (6.7e5)	**0.26** (**0.16**)	1.01 (0.51)	1.4e3 (1.1e4)	98.5 (501)	320 (1.9e3)
0.10	**1.13** (**0.98**)	1.72 (1.21)	1.9e3 (1.1e4)	51.5 (140)	182 (814)	**0.55** (**0.36**)	1.14 (0.59)	1.7e6 (1.7e7)	2.4e4 (2.2e5)	2.7e5 (2.7e6)	**0.27** (**0.16**)	0.97 (0.51)	5.4e3 (4.3e4)	315 (1.8e3)	1.1e3 (7.2e3)
0.15	**1.13** (**0.97**)	1.67 (1.08)	4.2e3 (2.4e4)	101 (274)	383 (1.8e3)	**0.59** (**0.47**)	1.16 (0.50)	3.8e6 (3.8e7)	5.5e4 (4.8e5)	6.0e5 (6.0e6)	**0.27** (**0.16**)	0.91 (0.41)	1.2e4 (9.6e4)	664 (4.0e3)	2.5e3 (1.6e4)
Clean	1.09 (1.05)	1.60 (1.10)	**0.82** (**0.88**)	0.84 (0.87)	1.62 (0.86)	0.53 (0.37)	1.08 (0.46)	**0.36** (**0.36**)	0.38 (0.36)	1.17 (0.35)	0.26 (0.16)	0.82 (0.39)	**0.17** (**0.15**)	0.18 (0.15)	0.99 (0.14)

of pixels N. Although the weight updates involve all n subjects, it exerts only a mild effect on computational time when $|\widetilde{\mathcal{M}}|$ is held fixed. Finally, our tests on a high-performance computing system equipped with 4.00 GB of RAM confirms that even the dense matrix $\mathbf{D}$ stays well within practical memory limits.

6.2. Simulation Study on the RSCC Procedure

The performance of the proposed robust SCC (RSCC) for the mean function is systematically evaluated and compared with the classical non-robust SCC (NRSCC) introduced by Wang et al. (2020). Following the model specifications and error distributions outlined at the beginning of Section 6, we simulate samples of size $n = 100,\ 200$ and 400. Empirical performance is assessed through two primary metrics: empirical coverage rates and average corridor widths, with the RSCC method benchmarked against the NRSCC approach.

Both SCC methods use 1,000 bootstrap replications, and compute empirical 95% SCC coverage from 100 simulation samples per scenario. We examine three contamination settings: (1) stripe and square contamination at 5% and 10% rates, (2) heavy-tailed error distributions

with mixing weights of 5% and 15%, and (3) uncontaminated samples as a baseline. Both the mean function estimation and the weighted bootstrap procedures use the Huber loss function with the tuning parameter $k = 0.1$. In the most challenging cases–specifically, stripe and square contamination at a 10% rate for all mean functions and heavy-tailed errors in the sine function setting–we apply $k = 0.01$ to ensure the robustness of our proposed RSCC procedure. For coverage assessment, we exclude the two outermost layers of pixels in the domain.

The simulation results for the RSCC and NRSCC are summarized in Table 3. For clean datasets, both the NRSCC and RSCC achieve comparable empirical coverage rates, with the RSCC exhibiting a slightly larger average corridor width. This is due to the loss of efficiency with the choice of a conservative Huber constant. Notably, as the sample size increases, the difference in corridor widths between the two methods diminishes. When contamination is introduced, the performance of the NRSCC begins to decline significantly.

In cases of surface-type contamination (stripe or square), our simulations show that although the NRSCC procedure can occasionally capture the true mean function with small sample sizes due to the inherent randomness, its empirical coverage declines significantly—even at a 5% contamination rate with $n = 100$. As the contamination rate or total number of images increases, the proportion of contaminated images rises, leading the NRSCC to consistently fail in achieving meaningful coverage. In contrast, the RSCC method maintains nominal coverage rates with stable corridor widths across most scenarios, demonstrating a clear edge. Similarly, under heavy-tailed error distributions, the RSCC attains nominal-level coverage without a substantial increase in corridor width, while the NRSCC produces excessively wide corridors that obscure rather than clarify the underlying signal. Finally, in the challenging sine function case with sharp local variations, both methods face inference difficulties; however, with an appropriate choice of k, the RSCC consistently achieves nominal coverage, whereas the NRSCC fails to do so. Overall, these simulations demonstrate that the RSCC method achieves reliable coverage rates and is far less sensitive to outlier images than the NRSCC

Table 3: Coverage rates (average corridor width) of the robust and non-robust versions of the 95% SCCs (RSCC and NRSCC) for all simulation settings. Bold values indicate the better-performing method in each scenario.

		Cont. Rate	$n = 100$		$n = 200$		$n = 400$	
Func. Type	Cont. Type	(Mix Weight)	RSCC	NRSCC	RSCC	NRSCC	RSCC	NRSCC
Quadratic	Stripe	0.05	**0.95 (0.829)**	0.50 (0.716)	**0.93 (0.480)**	0.09 (0.509)	**0.96 (0.329)**	0.00 (0.360)
		0.10	**0.96 (1.047)**	0.00 (0.782)	**0.96 (0.609)**	0.00 (0.556)	**0.95 (0.404)**	0.00 (0.394)
	Square	0.05	**0.94 (0.808)**	0.00 (0.662)	**0.92 (0.486)**	0.00 (0.471)	**0.95 (0.328)**	0.00 (0.335)
		0.10	**0.97 (1.138)**	0.00 (0.709)	**0.97 (0.610)**	0.00 (0.504)	**0.93 (0.405)**	0.00 (0.358)
	Cauchy	0.05	**0.93 (0.840)**	0.99 (4.166)	**0.91 (0.467)**	0.98 (1.709)	**0.96 (0.329)**	0.96 (1.725)
		0.15	**0.88 (0.754)**	0.95 (11.66)	**0.91 (0.461)**	0.95 (4.552)	**0.94 (0.332)**	0.96 (4.759)
	Slash	0.05	**0.93 (0.835)**	0.97 (1.768)	**0.91 (0.473)**	0.98 (3.691)	**0.95 (0.327)**	0.96 (3.314)
		0.15	**0.85 (0.708)**	0.94 (4.388)	**0.90 (0.467)**	0.97 (10.47)	**0.95 (0.335)**	0.97 (9.543)
	Clean	0.00	0.95 (0.809)	**0.93 (0.522)**	0.93 (0.479)	**0.97 (0.386)**	0.96 (0.329)	**0.97 (0.275)**
Exponential	Stripe	0.05	**0.90 (0.762)**	0.50 (0.716)	**0.94 (0.517)**	0.09 (0.509)	**0.94 (0.356)**	0.00 (0.360)
		0.10	**0.94 (1.389)**	0.00 (0.782)	**0.98 (0.831)**	0.00 (0.556)	**0.94 (0.525)**	0.00 (0.394)
	Square	0.05	**0.90 (0.760)**	0.00 (0.662)	**0.92 (0.517)**	0.00 (0.471)	**0.93 (0.355)**	0.00 (0.335)
		0.10	**0.94 (1.396)**	0.00 (0.709)	**0.97 (0.858)**	0.00 (0.504)	**0.90 (0.523)**	0.00 (0.358)
	Cauchy	0.05	**0.90 (0.763)**	0.77 (4.219)	**0.94 (0.514)**	0.70 (1.741)	**0.97 (0.363)**	0.58 (1.787)
		0.15	**0.89 (0.780)**	0.78 (11.74)	**0.94 (0.524)**	0.73 (4.625)	**0.94 (0.385)**	0.68 (4.935)
	Slash	0.05	**0.90 (0.756)**	0.78 (1.799)	**0.94 (0.515)**	0.72 (3.690)	**0.97 (0.361)**	0.54 (3.324)
		0.15	**0.91 (0.758)**	0.71 (4.491)	**0.94 (0.539)**	0.70 (10.55)	**0.94 (0.388)**	0.68 (9.571)
	Clean	0.00	0.90 (0.761)	**0.94 (0.541)**	0.95 (0.513)	**0.97 (0.386)**	0.98 (0.354)	**0.97 (0.275)**
Cubic	Stripe	0.05	**0.90 (0.683)**	0.50 (0.716)	**0.95 (0.471)**	0.09 (0.509)	**0.97 (0.338)**	0.00 (0.360)
		0.10	**0.94 (0.918)**	0.00 (0.782)	**0.96 (0.601)**	0.00 (0.556)	**0.98 (0.434)**	0.00 (0.394)
	Square	0.05	**0.90 (0.680)**	0.00 (0.662)	**0.95 (0.471)**	0.00 (0.471)	**0.97 (0.334)**	0.00 (0.335)
		0.10	**0.92 (0.914)**	0.00 (0.709)	**0.98 (0.606)**	0.00 (0.504)	**0.96 (0.423)**	0.00 (0.358)
	Cauchy	0.05	**0.90 (0.683)**	1.00 (4.105)	**0.96 (0.475)**	0.97 (1.648)	**1.00 (0.414)**	0.97 (1.658)
		0.15	**0.91 (0.695)**	0.99 (11.57)	**0.94 (0.507)**	0.99 (4.459)	**0.97 (0.372)**	0.99 (4.665)
	Slash	0.05	**0.91 (0.678)**	0.99 (1.671)	**0.94 (0.478)**	0.98 (3.602)	**1.00 (0.425)**	0.97 (3.243)
		0.15	**0.92 (0.680)**	0.97 (4.285)	**0.96 (0.542)**	1.00 (10.31)	**0.95 (0.343)**	1.00 (9.442)
	Clean	0.00	0.90 (0.683)	**0.94 (0.519)**	0.95 (0.468)	**0.96 (0.371)**	**0.99 (0.340)**	**0.97 (0.264)**
Sine	Stripe	0.05	**0.87 (0.918)**	0.50 (0.717)	**0.91 (0.593)**	0.06 (0.510)	**0.93 (0.397)**	0.00 (0.361)
		0.10	**0.93 (2.675)**	0.00 (0.783)	**0.95 (1.400)**	0.00 (0.556)	**0.93 (0.753)**	0.00 (0.394)
	Square	0.05	**0.86 (0.913)**	0.00 (0.719)	**0.92 (0.594)**	0.00 (0.511)	**0.94 (0.396)**	0.00 (0.363)
		0.10	**0.94 (2.858)**	0.00 (0.709)	**0.97 (1.392)**	0.00 (0.504)	**0.91 (0.742)**	0.00 (0.358)
	Cauchy	0.05	**0.95 (2.662)**	0.36 (4.463)	**0.98 (1.372)**	0.27 (1.963)	**0.98 (0.763)**	0.26 (2.018)
		0.15	**0.95 (2.670)**	0.47 (12.31)	**0.97 (1.354)**	0.38 (5.091)	**0.98 (0.789)**	0.27 (5.311)
	Slash	0.05	**0.96 (2.650)**	0.30 (2.103)	**0.95 (1.375)**	0.17 (4.001)	**0.99 (0.764)**	0.19 (3.605)
		0.15	**0.96 (2.671)**	0.35 (5.078)	**0.90 (1.366)**	0.47 (11.23)	**0.97 (0.798)**	0.43 (10.16)
	Clean	0.00	0.84 (0.915)	**0.94 (0.567)**	0.94 (0.589)	**0.97 (0.404)**	0.97 (0.395)	**0.97 (0.287)**

method, highlighting its robustness and reliability.

Additional simulation results, presented in Section S1.1 of the Supplementary Material, evaluate the performance of various choices for the Huber constant k. The findings indicate that reducing k from 0.1 to 0.01 improves the empirical coverage rate in scenarios with severe contamination, such as datasets containing 400 images following an exponential mean function, with 10% affected by square-type contamination. However, this smaller k also increases the average corridor width, underscoring that k should be selected based on the expected severity of contamination in the dataset.

In addition, separate simulations are performed to examine the effects of different triangulations ($\triangle$) and polynomial degrees d; see Section S1.1 of the Supplementary Material. The results demonstrate that the SCC is insensitive across these tuning parameters, provided the triangulation is sufficiently refined and the polynomial degree is not excessively low.

7. Applications

To demonstrate the proposed robust mean function estimation and simultaneous inference procedure, we use spatially normalized fludeoxyglucose PET imaging data from the Alzheimer's Disease Neuroimaging Initiative (ADNI), obtained from the ADNI database (`http://adni.loni.usc.edu`). The dataset includes PET images from 84 cognitively normal (CN) subjects, with each subject's scan comprising 68 horizontal slices at a resolution of 79×95 pixels per slice. In this section, we focus on the 20-th, 30-th, 40-th, and 50-th slices. The dataset reveals outlier images that exhibit patterns that deviate significantly from the mean, particularly in some boundary regions with unusually high or low pixel values. Figure 4 (a) displays the mean image for each slice, alongside examples of these outlier images.

We implement mean function estimation following the framework described in Section 5, with BPST parameters $d = 5$ and $r = 1$, employing weighted GCV for penalty parameter selection. We utilize the triangulation depicted in Figure 3 and set the Huber loss function constant to $k = 0.1$. Figure 4 (b) displays the mean function estimation along with the 95% SCC obtained by the proposed robust procedure for each of the four selected slices.

8. Conclusion and Discussions

In this paper, we develop a comprehensive statistical framework for robust image analysis that addresses two critical challenges: (1) the estimation of mean signal in a complex domain and (2) the detection and localization of significant effects, in the presence of heterogeneous noise patterns and outlier contamination. The methodological developments presented in this paper improve the reliability of statistical imaging inference, with broad applications

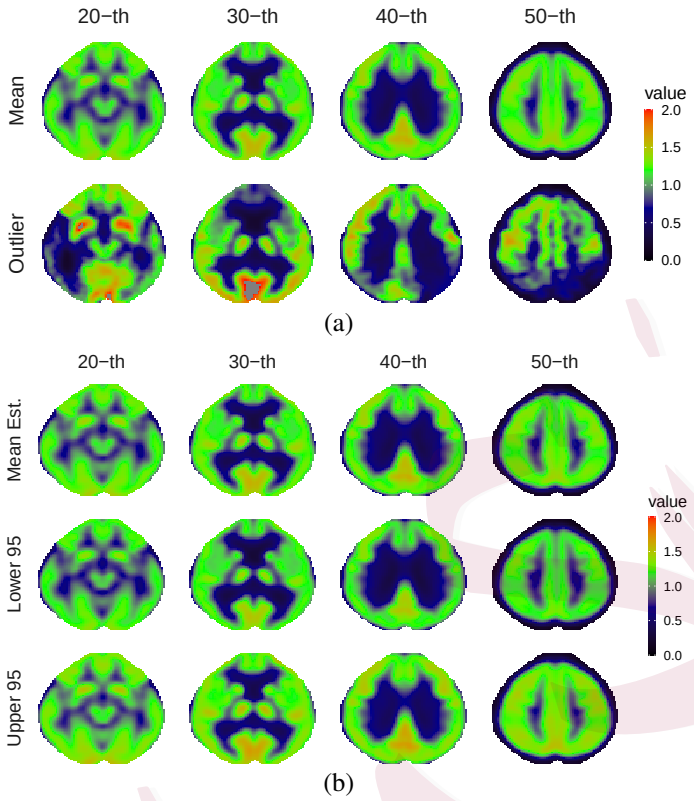


Figure 4: (a) an illustration of the sample average of the image slices of the CN subjects and examples of outlier images; (b) robust estimation and 95% RSCC of the mean image for the four selected slices.

in medical imaging-based research where data quality is often compromised by acquisition artifacts, physiological noise, and subject motion.

The proposed robust estimation and simultaneous inference procedure effectively handles heterogeneous noise structures and outlier contamination while preserving anatomically meaningful signals. The M-BPST estimator provides reliable mean function estimation even in the presence of outlier images or heavy-tailed errors. Our theoretical analysis establishes that the estimator is asymptotically consistent and achieves the optimal convergence rate under certain regularity conditions. We derive the asymptotic normality, supporting the construction of both pointwise confidence intervals and SCCs. Specifically, we employ a weighted bootstrap method to find a multiplicative factor for obtaining the robust SCC by expanding the pointwise variances.

Empirical studies demonstrate that the estimator performs competitively against classic

methods when no contamination is present and achieves comparable or superior performance to the RDNN-based mean function estimator, with substantially lower computational costs. The estimator exhibits particularly pronounced advantages over RDNN methods when analyzing irregularly shaped domains. Furthermore, our RSCC procedure provides reliable inference when outlier images are present or when errors have heavy-tailed distributions.

A natural extension of our methodology is to three-dimensional (3D) imaging data, which would substantially expand its applicability to volumetric neuroimaging analyses (Wang et al., 2025). This extension presents two primary challenges. First, the asymptotic properties and convergence rates of the M-BPST estimator would need to be re-established under increased dimensionality, potentially requiring new regularity conditions to account for the spatial complexity of 3D structures (Li et al., 2024). Second, scaling to 3D poses additional computational challenges. Specifically, the inherently larger number of tetrahedra required for volumetric partitioning leads to a significant increase in the number of basis functions. As a result, the construction of SCC in 3D space would necessitate novel computational strategies to efficiently manage the increased multiplicity of voxel-wise comparisons. In this context, parallelization, such as the domain decomposition discussed in Yu et al. (2025), and the use of specialized sparse data structures become crucial for both basis construction and optimization, while incorporating specialized bootstrap procedures optimized for 3D spatial dependencies is essential to maintain computational feasibility in whole-brain analyses.

Another promising direction for future research is the data-driven selection of the Huber constant. While traditional approaches for homogeneous errors typically involve scaling residuals using a pre-calculated robust scale estimator, the extension to heterogeneous errors demands more sophisticated methodology. One promising direction involves adapting the methodology to explicitly handle heterogeneity. For example, instead of relying on a global scale estimator, localized or pixel-specific scale estimators could be considered, though this requires careful consideration of the robustness-efficiency trade-off. Building upon the

adaptive Huber regression framework of Sun et al. (2020), an alternative approach could involve developing sophisticated adaptation mechanisms for the Huber constant that respond to both local error characteristics and problem complexity. Such an approach would necessitate careful modification of existing adaptive methods to accommodate heterogeneous error assumptions, potentially through the integration of spatially-varying penalty parameters calibrated to local error scales and structural features of the imaging data.

Supplementary Material

In the Supplementary Material, we provide additional simulation studies and detailed proofs of the theoretical results in this paper.

Acknowledgement

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`http://adni.loni.usc.edu`). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: `http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf`.

The research work of Yang Long and Lily Wang is partially supported by the National Institutes of Health award 1R01AG085616 and the National Science Foundation award 2426173. Guanqun Cao's research is partially supported by the National Science Foundation under Grants DMS-2413301, CNS-2319342 and CNS-2319343.

This project was further supported by resources provided by the Office of Research Computing at George Mason University (URL: `https://orc.gmu.edu`) and funded in part by grants from the National Science Foundation (Award Number 2018631).

References

Bali, J. L., Boente, G., Tyler, D. E., and Wang, J.-L. (2011), "Robust functional principal components: a projection-pursuit approach," *The Annals of Statistics*, 39, 2852–2882.

Belloni, A., Chernozhukov, V., Chetverikov, D., and Fernández-Val, I. (2019), "Conditional quantile processes based on series or many regressors," *Journal of Econometrics*, 213, 4–29, annals: In Honor of Roger Koenker.

Boente, G. and Salibian-Barrera, M. (2015), "S-Estimators for functional principal component analysis," *Journal of the American Statistical Association*, 110, 1100–1111.

Chang, C., Lin, X., and Ogden, R. T. (2017), "Simultaneous confidence bands for functional regression models," *Journal of Statistical Planning and Inference*, 188, 67–81.

Chen, X. and Zhou, W.-X. (2020), "Robust inference via multiplier bootstrap," *The Annals of Statistics*, 48, 1665–1691.

Choi, H. and Reimherr, M. (2017), "A geometric approach to confidence regions and bands for functional parameters," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80, 239–260.

Degras, D. (2017), "Simultaneous confidence bands for the mean of functional data," *Wiley Interdisciplinary Reviews: Computational Statistics*, 9, e1397.

Eddelbuettel, D. and Sanderson, C. (2014), "RcppArmadillo: Accelerating R with high-performance C++ linear algebra," *Computational Statistics & Data Analysis*, 71, 1054–1063.

Gervini, D. (2008), "Robust functional estimation using the median and spherical principal components," *Biometrika*, 95, 587–600.

Gu, L., Wang, L., Härdle, W. K., and Yang, L. (2014), "A simultaneous confidence corridor for varying coefficient regression with sparse functional data," *TEST*, 23, 806–843.

Gu, Z., Yu, S., Wang, G., Lai, M.-J., and and, L. W. (2025), "TSSS: a novel triangulated spherical spline smoothing for surface-based data," *Journal of Nonparametric Statistics*, 0, 1–30.

Huang, C. and Zhu, H. (2022), "Functional hybrid factor regression model for handling heterogeneity in imaging studies," *Biometrika*, 109, 1133–1148.

Huber, P. J. (1964), "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, 35, 73–101.

— (1973), "Robust regression: asymptotics, conjectures and Monte Carlo," *The Annals of Statistics*, 1, 799–821.

Kalogridis, I. and Van Aelst, S. (2023a), "Robust optimal estimation of location from discretely

sampled functional data," *Scandinavian Journal of Statistics*, 50, 411–451.

— (2023b), "Robust penalized estimators for functional linear regression," *Journal of Multivariate Analysis*, 194, 105104.

Lai, M.-J. and Schumaker, L. L. (2007), *Spline Functions on Triangulations*, Encyclopedia of Mathematics and Its Applications, Cambridge: Cambridge University Press.

Lai, M.-J. and Wang, L. (2013), "Bivariate penalized splines for regression," *Statistica Sinica*, 23, 1399–1417.

Lee, S., Shin, H., and Billor, N. (2013), "M-type smoothing spline estimators for principal functions," *Computational Statistics & Data Analysis*, 66, 89–100.

Li, J. and Yang, L. (2023), "Statistical inference for functional time series," *Statistica Sinica*, 33, 519–549.

Li, X., Yu, S., Wang, Y., Wang, G., Wang, L., and Lai, M.-J. (2024), "Nonparametric regression for 3D point cloud learning," *Journal of Machine Learning Research*, 25, 1–56.

Lima, I. R., Cao, G., and Billor, N. (2019a), "M-based simultaneous inference for the mean function of functional data," *Annals of the Institute of Statistical Mathematics*, 71, 577–598.

— (2019b), "Robust simultaneous inference for the mean function of functional data," *TEST*, 28, 785–803.

Liu, Y., Li, M., and Morris, J. S. (2023), *Scalable function-on-scalar quantile regression for densely sampled functional data*, available at: `https://arxiv.org/abs/2002.03355`.

Maronna, R. A., Martin, R. D., Yohai, V. J., and Salibián-Barrera, M. (2019), *Robust Statistics: Theory and Methods (with R)*, Wiley Series in Probability and Statistics, Hoboken, NJ: John Wiley & Sons, Ltd, 2nd ed.

Sanderson, C. and Curtin, R. (2016), "Armadillo: a template-based C++ library for linear algebra," *Journal of Open Source Software*, 1, 26.

Shewchuk, J. R. (1994), "An introduction to the conjugate gradient method without the agonizing pain," Tech. rep., Carnegie Mellon University, USA.

Shieh, D. and Ogden, R. T. (2023), "Permutation-based inference for function-on-scalar regression with an application in PET brain imaging," *Journal of Nonparametric Statistics*, 35, 820–838.

Shin, H. and Lee, S. (2016), "An RKHS approach to robust functional linear regression," *Statistica Sinica*, 26, 255–272.

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

Sun, Q., Zhou, W.-X., and Fan, J. (2020), "Adaptive Huber regression," *Journal of the*

American Statistical Association, 115, 254–265.

Tang, Q. and Cheng, L. (2008), "M-estimation and B-spline approximation for varying coefficient models with longitudinal data," *Journal of Nonparametric Statistics*, 20, 611–625.

Wang, S. and Cao, G. (2022), "Robust deep neural network estimation for multi-dimensional functional data," *Electronic Journal of Statistics*, 16, 6461–6488.

Wang, Y., Wang, G., Klinedinst, B., Willette, A., and Wang, L. (2025), "Statistical inference for mean functions of complex 3D objects," *Statistica Sinica*, 35, 1–27.

Wang, Y., Wang, G., Wang, L., and Ogden, R. T. (2020), "Simultaneous confidence corridors for mean functions in functional data analysis of imaging data," *Biometrics*, 76, 427–437.

Wei, Y. and He, X. (2006), "Conditional growth charts," *The Annals of Statistics*, 34, 2069–2097.

Wood, S. (2017), *Generalized Additive Models: An Introduction with R*, Chapman and Hall/CRC, 2nd ed.

Wood, S. N. (2003), "Thin plate regression splines," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65, 95–114.

Wood, S. N., Bravington, M. V., and Hedley, S. L. (2008), "Soap film smoothing," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70, 931–955.

Yu, S., Wang, G., and Wang, L. (2025), "Distributed heterogeneity learning for generalized partially linear models with spatially varying coefficients," *Journal of the American Statistical Association*, 120, 779–793.

Yu, S., Wang, G., Wang, L., Liu, C., and Yang, L. (2020), "Estimation and inference for generalized geoadditive models," *Journal of the American Statistical Association*, 115, 761–774.

Yu, S., Wang, G., Wang, L., and Yang, L. (2021), "Multivariate spline estimation and inference for image-on-scalar regression," *Statistica Sinica*, 31, 1463–1487.

Zhong, C. and Yang, L. (2023), "Statistical inference for functional time series: autocovariance function," *Statistica Sinica*, 33, 2519–2543.

Zhou, X., Kong, D., Kashlak, A., Kong, L., Karunamuni, R., and Zhu, H. (2023), "Functional response quantile regression model," *Statistica Sinica*, 33, 2643–2667.

Zhu, H., Fan, J., and Kong, L. (2014), "Spatially varying coefficient model for neuroimaging data With jump discontinuities," *Journal of the American Statistical Association*, 109, 1084–1098.