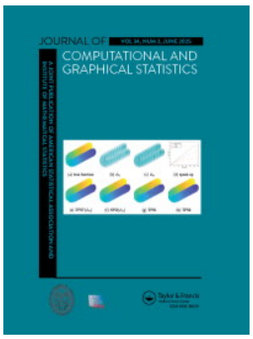


Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: www.tandfonline.com/journals/ucgs20

Efficient Nonparametric Estimation of 3D Point Cloud Signals through Distributed Learning

Guannan Wang, Yuchun Wang, Annie S. Gao, Li Wang & Alzheimer's Disease Neuroimaging Initiative

To cite this article: Guannan Wang, Yuchun Wang, Annie S. Gao, Li Wang & Alzheimer's Disease Neuroimaging Initiative (2025) Efficient Nonparametric Estimation of 3D Point Cloud Signals through Distributed Learning, Journal of Computational and Graphical Statistics, 34:2, 746-758, DOI: 10.1080/10618600.2024.2406301

To link to this article: https://doi.org/10.1080/10618600.2024.2406301

View supplementary material

Published online: 18 Nov 2024.

Submit your article to this journal

Article views: 160

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=ucgs20

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2025, VOL. 34, NO. 2, 746–758
https://doi.org/10.1080/10618600.2024.2406301

Taylor & Francis
Taylor & Francis Group

Check for updates

Efficient Nonparametric Estimation of 3D Point Cloud Signals through Distributed Learning

Guannan Wang[a], Yuchun Wang[b], Annie S. Gao[c], Li Wang[b], and Alzheimer's Disease Neuroimaging Initiative

[a]Department of Mathematics, William & Mary, Williamsburg, VA; [b]Department of Statistics, George Mason University, Fairfax, VA; [c]McLean High School, McLean, VA

ABSTRACT

Advancements in technology have elevated the prominence of 3D point cloud data, making its analysis increasingly vital across various applications. This need drives the demand for advanced statistical analytic approaches to handle challenges such as size, sparsity, and irregularity in 3D point clouds while ensuring accurate and efficient information extraction. This article introduces a novel nonparametric distributed (NPD) learning framework that uses trivariate spline smoothing over a triangulation of domain. The proposed NPD algorithm features a straightforward, scalable, and communication-efficient implementation scheme that can achieve near-linear speedup. In addition, we provide rigorous theoretical support for the NPD estimation framework and demonstrate that the NPD spline estimators attain the same convergence rate as the global spline estimators obtained using the entire dataset and achieve the optimal nonparametric convergence rate established by Stone under some regularity conditions. To evaluate the efficacy of the proposed NPD method, we conduct simulation studies comparing it with several global nonparametric estimation methods used to smooth the 3D data. The results demonstrate the superior performance of the NPD method in accurately and efficiently processing and learning from 3D point clouds, highlighting its potential to advance large and complex data analysis. Supplementary material, which contains related technical details, proofs of the theoretical results, and additional results in simulation studies, is available online.

ARTICLE HISTORY
Received February 2024
Accepted September 2024

KEYWORDS
Complex 3D data;
Computational efficiency;
Distributed learning;
Nonparametric smoothing;
Trivariate spline smoothing

1. Introduction

In recent years, the prevalence of three-dimensional (3D) point cloud data has expanded across various fields, driven by the increasing accessibility of advanced data acquisition technologies and the growing demand for highly detailed 3D information. These data provide rich insights into the structure, geometry, and characteristics of real-world objects and environments, making them invaluable resources for various applications and industries. Efficiently extracting meaningful information from 3D point cloud data has become increasingly important across multiple applications.

Beyond representing the shape of objects, point cloud data often encompass additional attributes that carry important real-life information, such as reflectance, color, or other relevant properties. Applications such as robotics, autonomous vehicles, virtual and augmented reality, 3D modeling, and medical imaging rely heavily on effectively extracting signals from 3D point cloud data to achieve their desired outcomes. For example, in medical imaging, 3D point cloud data derived from techniques such as computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET) can be used to visualize and analyze complex anatomical structures and pathologies. Accurate extraction and interpretation of signals from medical images are essential for informed clinical decision-making, aiding in diagnosis and treatment planning. Similarly, autonomous vehicles require precise and real-time object detection and segmentation, which consider not only geometry but also other properties, such as reflectance, to navigate safely. By leveraging the rich information contained in 3D point cloud data, autonomous systems can make informed decisions and respond effectively to dynamic situations. However, the processing and learning of 3D point cloud data present unique challenges that require careful consideration to extract valuable information accurately and efficiently.

First, the high dimensionality of point cloud data often results in a massive number of points, making it computationally expensive to process and analyze. This challenge is particularly pronounced when dealing with real-time or large-scale applications, where timely processing is essential. In medical imaging, the enormous size of point cloud data can be illustrated through volumetric datasets obtained from techniques like CT, MRI, or PET. These imaging modalities generate detailed 3D representations of internal structures, capturing not only the surfaces of anatomical structures or pathologies but also the distribution of points throughout the volume of the scanned region. As a result, these volumetric point clouds can encompass

CONTACT Li Wang lwang41@gmu.edu Department of Statistics, George Mason University, 1721 Nguyen Engineering Building, 4400 University Drive, MS 4A7, Fairfax, VA 22030.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

© 2024 American Statistical Association and Institute of Mathematical Statistics

millions of points, representing various tissues, organs, and abnormalities with a high degree of precision. For example, a standard PET scan of one subject contains $(79 \times 95 \times 68)$ 510,340 voxels. Consequently, even for a single image, the number of observations is considerably high, sometimes beyond the storage capability of a single computer, highlighting the challenges associated with processing and analyzing such high-dimensional data.

To handle the massive volume of data, many developments in distributed and communication-efficient statistical methods have recently been made in a broad spectrum of topics, such as generalized linear regression (Wang, Zhu, and Ma 2018; Meng et al. 2021), M-estimation (Battey et al. 2018; Shi, Lu, and Song 2018; Banerjee, Durot and Sen 2019), density estimation (Zhang et al. 2023), nonparametric regression (Shang and Cheng 2017; Szabó and van Zanten 2019; Yu, Wang, and Wang 2024), quantile regression (Volgushev, Chao, and Cheng 2019; Chen, Liu, and Zhang 2022), spatiotemporal regression (Wang, Wang, and Gao 2024), bootstrap (Kleiner et al. 2014), and Bayesian methods (Song and Liang 2014; Jordan, Lee, and Yang 2019). Most of these distributed learning methods are closely related to the idea of divide-and-conquer, which breaks down a problem into many sub-problems of the same or similar type and then combines the solutions of these sub-problems to solve the original problem. The primary strategy in the divide step is the subsampling approach. Specifically, subsamples obscuring the underlying signals' features are obtained based on each subsample; then, these estimators are combined through a simple average, one-shot approach, and adaptive method. Excellent reviews and discussions of these distributed learning methods, inferences, and software are presented in Wang et al. (2016) and Gao et al. (2022). Furthermore, most existing methods focus on solving the big data challenge under the linear regression framework and cannot be directly applied to 3D point cloud analysis.

Second, 3D point clouds usually contain unwanted noise that obscures the features of the underlying signals. Consequently, denoising is critical in the analysis; for example, in many imaging studies, smoothing methods are widely adopted in image processing tasks (Goldsmith, Huang, and Crainiceanu 2014; Wong et al. 2016; Kang 2020; Wang et al. 2021). Additionally, point clouds frequently exhibit irregular shapes or holes due to their adaptability to capture and analyze complex object and environment geometry. For example, in medical imaging, techniques such as MRI and CT generate point clouds that faithfully depict anatomical structures. Organs and bones have intricate shapes, and regions like blood vessels or hollow organs can cause voids in the data. Similarly, in robotics and artificial intelligence, point clouds facilitate environment perception. Objects encountered by robots often exhibit irregular shapes, making traditional geometric modeling challenging.

Another significant challenge in processing and learning from point cloud data is data sparsity, a common occurrence resulting from various technical, environmental, and intrinsic factors affecting data acquisition and processing. This data scarcity becomes particularly pronounced in inaccessible regions, such as those with occlusions, holes, cracks, and slits. As a result, it is crucial to devise innovative techniques capable of adeptly managing the irregularities and sparsity inherent in the data while maintaining the essential information needed for accurate feature extraction and pattern recognition.

Traditional smoothing techniques often struggle when applied to point clouds with irregular shapes and holes. Many conventional denoising methods, such as kernel smoothing (Zhu, Fan, and Kong 2014), tensor product smoothing (Reiss and Ogden 2010), thin plate spline smoothing (Yue and Speckman 2010), and wavelet smoothing (Morris and Carroll 2006), suffer from the problem of "leakage" across complex domains, which refers to poor estimation over difficult regions caused by smoothing inappropriately across boundary features (Ramsay 2002; Wang and Ranalli 2007; Kim and Wang 2021; Jiang et al. 2021). Several techniques have been developed in the last two decades to tackle the "leakage" problem when recovering the underlying signals. Much of the work focuses on 2D manifold domains, such as spline smoothing (Wilhelm et al. 2016), finite element analysis (Lindgren et al. 2011), and kernel smoothing (Guo, Duan, and Shum 2010). For 3D cases, Chung, Wang, and Wu (2018) applied the discrete heat kernel smoothing to irregular image domains and Huo et al. (2020) considered the finite element method for irregular 3D subjects.

In this article, we address these challenges by introducing a novel nonparametric distributed (NPD) method designed to effectively extract signals from large and complex 3D point clouds. To provide a framework for our work, we consider a point cloud comprising n points within a specified domain $\Omega \subset \mathbb{R}^3$, which may exhibit irregularities with complex boundaries and include holes. The location or design point of the ith observation is denoted as $\mathbf{X}_i$, where $i = 1, \ldots, n$. Each $\mathbf{X}_i = (X_{i1}, X_{i2}, X_{i3})$ represents a distinct 3D spatial coordinate in the cloud. Correspondingly, let Y_i denote the response variable observed at location $\mathbf{X}_i$. Then, we regard any point cloud as a set of n observations $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ in general, and assume that these observations conform to the following nonparametric regression model:

$$Y_i = m(\mathbf{X}_i) + \sigma(\mathbf{X}_i)\varepsilon_i, \ \mathbf{X}_i \in \Omega, \tag{1}$$

where $m(\cdot)$ represents the unknown conditional mean function capturing the underlying signal, $\sigma(\cdot)$ denotes the unknown conditional standard deviation function, and ε_i denotes the random error term with zero mean. Our primary interest is estimating the mean function $m(\cdot)$, which characterizes the essential information within the point cloud data.

One particularly relevant and powerful technique for processing and learning from point cloud data is multivariate spline smoothing over a triangulation of the domain, which enables efficient approximation of complex, irregular data structures by constructing triangulations that capture the underlying geometry and features of the point cloud. The trivariate penalized spline on triangulation (Li et al. 2024, TPST) is well suited to handle the sparsity and heterogeneity inherent in complex 3D data, providing a flexible solution for accurate feature extraction and pattern recognition. One key advantage of this nonparametric method lies in its computational simplicity. As a global estimation with an explicit model expression, the fitting of splines is achieved by solving a single linear system.

Advancing the TPST method, this article addresses the challenges posed by large point cloud datasets. We propose a novel

distributed learning approach that uses domain decomposition (DDC) to maintain the intricate spatial structure inherent in the data. This approach contrasts with the standard subsampling techniques and random division strategies used in the divide-and-conquer paradigm. Using the TPST method, we use triangulation to accurately approximate a given complex domain and partition it into a set of disjoint tetrahedra. Subsequently, spline basis functions are constructed on each tetrahedron. Unlike conventional spline methodologies, in which smoothness is determined by polynomial degrees, the global smoothness of spline functions within TPST is attained by introducing linear constraints on the spline coefficients. This feature can be easily guaranteed in the aggregation step at the end without extra communications among different worker processors. Communication efficiency is crucial in scenarios where data is distributed, especially in contexts with communication expenses or privacy considerations. The DDC strategy can effectively improve communication efficiency and load balancing, thereby streamlining the process.

Combining the strengths of distributed learning with the power of trivariate spline smoothing, our proposed NPD framework adeptly handles the challenges associated with 3D point cloud data, including its enormous size, irregularity, and sparsity. In addition, we offer rigorous theoretical support for the proposed NPD learning method. Specifically, we demonstrate that the NPD-based TPST estimators achieve the same convergence rate as the global TPST estimators derived from the entire dataset, thereby attaining the optimal convergence rate as demonstrated by Stone (1982) within the nonparametric regression framework. This result highlights the effectiveness of our approach in learning from large and complex 3D point cloud data.

The remainder of the article is structured as follows. In Section 2, we provide an overview of the trivariate spline over triangulation and introduce the trivariate penalized spline over triangulation (TPST) estimators. In Section 3, we introduce the distributed learning framework based on TPST. Section 3.1 details the NPD-based TPST algorithm, while Sections 3.2 and 3.3 present the asymptotic properties of the NPD-based TPST estimators and implementation details, respectively. We demonstrate the performance of both the global smoothers and the distributed learning method in numerical examples in Section 4. Finally, this article is concluded, and future work is described in Section 5. The supplementary material includes additional simulation results and in-depth theoretical details for further reference.

2. An Overview of Global Trivariate Spline Smoothing

In this section, we present an overview of the trivariate spline-based estimation of the mean function $m(\cdot)$ in (1) using the entire set of observations from the point cloud. First, we discuss the construction of triangulation from a 3D point cloud and provide a fundamental understanding of triangulations and trivariate penalized splines over triangulations. Subsequently, we introduce the global penalized spline estimators, which serve as a reference for the proposed NPD framework.

2.1. From Point Clouds to Triangulation

The domain of a point cloud, Ω, is often defined as a polygonal domain encompassing a geometric region composed of polygons. This versatile representation accommodates a wide range of various objects. In cases where the domain of a point cloud is unknown, surface reconstruction techniques such as triangular meshes can be employed, which enable the determination of a tight polygonal domain encompassing all points in the cloud.

Based on the polygonal domain, we can use the Delaunay triangulation algorithm to partition the domain into a collection of tetrahedra. A tetrahedron T refers to the convex hull of four noncoplanar points in $\mathbb{R}^3$. A collection $\triangle = \{T_1, \ldots, T_N\}$ of N tetrahedra is referred to as a triangulation of the domain Ω, satisfying the condition that any pair of tetrahedra in $\triangle$ intersect at most at a common vertex, along a common edge or a common triangular face. Various packages and toolboxes exist for practical purposes. For example, MATLAB provides functions such as `delaunayTriangulaion`, `delaunay` and `distmesh/distmeshnd` (Persson and Strang 2004), while C++ offers options such as `QualMesh` (Cheng et al. 2005) and the CGAL library (Jamin et al. 2015; The CGAL Project 2020). In addition, `TetGen`(Si 2015) and `iso2mesh` (Fang and Boas 2009), built on `CGAL`, are available for MATLAB/Octave.

2.2. Trivariate Splines Over Triangulation

Upon the construction of the domain triangulation, we proceed to establish the foundation for trivariate spline bases over this triangulation. We begin by introducing the trivariate Bernstein basis polynomials, followed by the definition of the trivariate spline space in the following.

Given a tetrahedron $T = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4 \rangle$, any fixed point $\boldsymbol{x} := (x_1, x_2, x_3) \in \mathbb{R}^3$ has a unique representation in terms of $\langle \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4 \rangle$, $\boldsymbol{x} = b_1\boldsymbol{v}_1 + b_2\boldsymbol{v}_2 + b_3\boldsymbol{v}_3 + b_4\boldsymbol{v}_4$, with $b_1 + b_2 + b_3 + b_4 = 1$, where (b_1, b_2, b_3, b_4) are called the *barycentric coordinates* of $\boldsymbol{x}$ *relative to the tetrahedron* T, and they are nonnegative if $\boldsymbol{x}$ is inside or on the faces of T. Accordingly, if $\boldsymbol{x} \in T$, for some nonnegative integers i, j, k, l with $i+j+k+l = d$, define *trivariate Bernstein basis polynomial of degree d relative to* T as $B^{d,T}_{ijkl}(\boldsymbol{x}) = d!/(i!j!k!l!) b_1^i b_2^j b_3^k b_4^l$, with $i + j + k + l = d$; if $\boldsymbol{x} \notin T$, $B^{d,T}_{ijkl}(\boldsymbol{x}) = 0$.

For any positive integer d and tetrahedron T, let $\mathcal{P}_d(T)$ be the space of all trivariate polynomials defined on T with degrees less than or equal to d. The dimension of this space, $\mathcal{P}_d(T)$, is $\binom{d+3}{3}$. The set of Bernstein basis polynomials $\boldsymbol{B}^d_T(\boldsymbol{x}) := \{B^{d,T}_{ijkl}(\boldsymbol{x}) : i, j, k, l \geq 0, i + j + k + l = d\}$ forms a basis for the space of polynomials $\mathcal{P}_d(T)$. Thus, given Bernstein basis functions $\{B^{d,T}_{ijkl}(\boldsymbol{x})\}_{i+j+k+l=d}$, any polynomial $p(\boldsymbol{x}) \in \mathcal{P}_d(T)$ can be written uniquely as *B-form*, $p(\boldsymbol{x})|_T = \sum_{i+j+k+l=d} \gamma_{T;ijkl} B^{d,T}_{ijkl}(\boldsymbol{x}) = \boldsymbol{B}^d_T(\boldsymbol{x})^\top \boldsymbol{\gamma}_T$, where the coefficients $\boldsymbol{\gamma}_T = \{\gamma_{T;ijkl}\}_{i+j+k+l=d}$ are called *B-coefficients* of p.

For domain Ω, let $\mathcal{C}^r(\Omega)$ be the collection of all rth continuously differentiable functions over Ω for some nonnegative integer r. For the tetrahedral partition $\triangle = \{T_1, \ldots, T_N\}$, let

$\mathcal{S}_d^r(\triangle) = \{s \in \mathcal{C}^r(\Omega) : s|_T \in \mathcal{P}_d(T), T \in \triangle\}$ be a *spline space of degree d and smoothness r over* $\triangle$, where $s|_T$ is the polynomial piece of spline s restricted in the tetrahedron T. For any $s \in \mathcal{S}_d^r(\triangle)$, there exists $\boldsymbol{\gamma}_{T_J} = \{\gamma_{T_J;ijkl}\}$ such that $s|_{T_J}(\boldsymbol{x}) = \sum_{i+j+k+l=d} \gamma_{T_J;ijkl} B_{ijkl}^{d,T_J}(\boldsymbol{x}) = \boldsymbol{B}_{T_J}^d(\boldsymbol{x})^\top \boldsymbol{\gamma}_{T_J}$, $J = 1, \ldots, N$. Consequently, letting $\boldsymbol{B}_d = ((\boldsymbol{B}_{T_1}^d)^\top, \ldots, (\boldsymbol{B}_{T_N}^d)^\top)^\top$ and $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_{T_1}^\top, \ldots, \boldsymbol{\gamma}_{T_N}^\top)^\top$, one has $s(\boldsymbol{x}) = \boldsymbol{B}_d(\boldsymbol{x})^\top \boldsymbol{\gamma}$, for any $s \in \mathcal{S}_d^r(\triangle)$.

2.3. Global Trivariate Penalized Spline Estimators

As mentioned above, point clouds often contain regions with sparse data. In such cases, penalized splines offer a convenient and effective tool for data fitting (Green and Silverman 1994; Wood 2003; Lai and Wang 2013). Penalized splines can adapt and flexibly capture the underlying structure and patterns in the data, even in regions with limited observations. By imposing penalties, the fitting process can balance achieving a good fit for the available data points and maintaining smoothness throughout the domain. This flexibility and adaptability make penalized splines well-suited for handling sparse data regions in point clouds and extracting meaningful information. In the following, we describe the global penalized spline estimator of the function $m(\cdot)$ in (1).

We approach the problem of penalized spline smoothing through the formulation of a penalized least squares objective:

$$\min_{s \in \mathcal{S}_d^r(\triangle)} \sum_{i=1}^n \{Y_i - s(\boldsymbol{x}_i)\}^2 + \lambda \mathcal{E}(s), \tag{2}$$

where $\mathcal{E}(\cdot)$ represents a penalty function and λ serves as the roughness penalty parameter. The appropriate choice of the penalty parameter λ strikes a balance between data fit goodness and the stability of estimated functions. To capture derivatives of order $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \alpha_3$, we introduce the notation $D^{\boldsymbol{\alpha}} s = \partial^{|\boldsymbol{\alpha}|} s / \partial_{x_1}^{\alpha_1} \partial_{x_2}^{\alpha_2} \partial_{x_3}^{\alpha_3}$. A common penalty function can be defined as

$$\mathcal{E}(s) = \sum_{T \in \triangle} \mathcal{E}(s_T) = \sum_{T \in \triangle} \sum_{\alpha_1+\alpha_2+\alpha_3=2} \binom{2}{\alpha_1} \binom{2-\alpha_1}{\alpha_2} \times \int_T \left\{ D_{x_1}^{\alpha_1} D_{x_2}^{\alpha_2} D_{x_3}^{\alpha_3} s_T(\boldsymbol{x}) \right\}^2 \mathrm{d}\boldsymbol{x},$$

where $s_T(\boldsymbol{x}) = \sum_{i+j+k+l=d} \gamma_{T;ijkl} B_{ijkl}^{T,d}(\boldsymbol{x})$ is a function restricted on T.

Defining $s(\boldsymbol{x}) = \mathbf{B}_d(\boldsymbol{x})^\top \boldsymbol{\gamma}$, we can express $\mathcal{E}(\mathbf{B}_d^\top \boldsymbol{\gamma})$ as $\boldsymbol{\gamma}^\top \mathbf{P} \boldsymbol{\gamma}$, where $\mathbf{P}$ represents the block diagonal penalty matrix. The *Trivariate Penalized Spline over Triangulation* (TPST) estimator of m, denoted as $\widehat{m}^{\mathrm{G}}(\boldsymbol{x}) \equiv \widehat{m}^{\mathrm{G}}(\boldsymbol{x}; \lambda) = \mathbf{B}_d(\boldsymbol{x})^\top \widehat{\boldsymbol{\gamma}}_\lambda^{\mathrm{G}}$, is obtained by minimizing a constrained objective function with respect to the spline coefficients $\widehat{\boldsymbol{\gamma}}_\lambda^{\mathrm{G}}$. The exact formulation of the objective function is provided below:

$$\sum_{i=1}^n \{Y_i - \mathbf{B}_d(\mathbf{X}_i)^\top \boldsymbol{\gamma}\}^2 + \lambda \boldsymbol{\gamma}^\top \mathbf{P} \boldsymbol{\gamma}, \text{ subject to } \mathbf{H}\boldsymbol{\gamma} = \mathbf{0}, \tag{3}$$

where $\mathbf{H}$ represents the matrix incorporating smoothness conditions across shared edges or faces of tetrahedra.

The QR decomposition technique can be employed to eliminate the constraint imposed on the problem. Applying QR decomposition, we obtain the factorization $\mathbf{H}^\top = \mathbf{QR} = (\mathbf{Q}_1\ \mathbf{Q}_2)\binom{\mathbf{R}_1}{\mathbf{R}_2}$, where $\mathbf{Q}$ represents an orthogonal matrix, $\mathbf{R}$ is an upper triangular matrix, $\mathbf{R}_1$ is a matrix of full rank with the same rank as $\mathbf{H}$, and $\mathbf{R}_2$ is a matrix composed entirely of zeros. In particular, for any vector $\boldsymbol{\gamma}$ that satisfies $\mathbf{H}\boldsymbol{\gamma} = \mathbf{0}$, there exists a corresponding vector $\boldsymbol{\theta}$ such that $\boldsymbol{\gamma} = \mathbf{Q}_2\boldsymbol{\theta}$. Moreover, it is worth noting that for any $\boldsymbol{\theta}$, condition $\mathbf{H}(\mathbf{Q}_2\boldsymbol{\theta}) = \mathbf{0}$ is satisfied. Hence, the original objective (3) can be transformed into an unconstrained penalized regression problem:

$$\sum_{i=1}^n \left\{ Y_i - \mathbf{B}_d(\mathbf{X}_i)^\top \mathbf{Q}_2 \boldsymbol{\theta} \right\}^2 + \lambda \boldsymbol{\theta}^\top \mathbf{Q}_2^\top \mathbf{P} \mathbf{Q}_2 \boldsymbol{\theta}. \tag{4}$$

Let $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$, $\widetilde{\mathbf{B}}_d = (\mathbf{B}_d(\mathbf{X}_1), \ldots, \mathbf{B}_d(\mathbf{X}_n))^\top$. Then, the minimizer of (4) can be obtained as $\widehat{\boldsymbol{\theta}}_\lambda^{\mathrm{G}} = (\mathbf{Q}_2^\top \widetilde{\mathbf{B}}_d^\top \widetilde{\mathbf{B}}_d \mathbf{Q}_2 + n^{-1}\lambda \mathbf{Q}_2^\top \mathbf{P} \mathbf{Q}_2)^{-1} \mathbf{Q}_2^\top \widetilde{\mathbf{B}}_d^\top \mathbf{Y}$ and $\widehat{\boldsymbol{\gamma}}_\lambda^{\mathrm{G}} = \mathbf{Q}_2 \widehat{\boldsymbol{\theta}}_\lambda^{\mathrm{G}}$, from which we derive the global TPST estimator of m: $\widehat{m}^{\mathrm{G}}(\boldsymbol{x}) = \mathbf{B}_d(\boldsymbol{x})^\top \widehat{\boldsymbol{\gamma}}_\lambda^{\mathrm{G}}$.

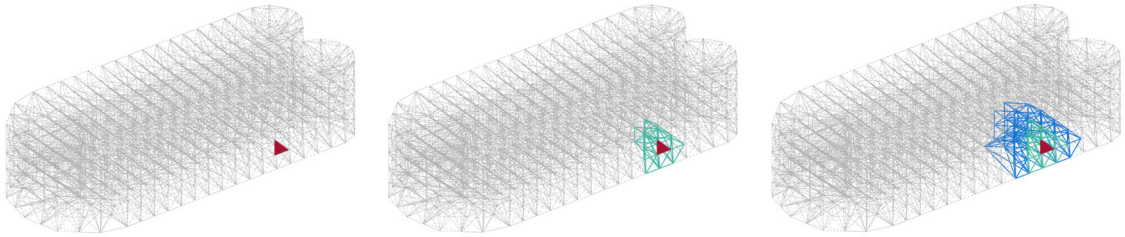
3. Nonparametric Distributed (NPD) Learning

This section introduces a novel approach to uncovering 3D point cloud signals using a distributed learning framework. In contrast to the global estimation method discussed in Section 2.3 that relies on the entire dataset, the distributed learning method leverages the power of distributed computing to process and analyze extraordinarily large-scale point cloud data efficiently. In the following, we outline the key concepts and methodology behind the NPD approach and highlight its advantages regarding computational and communicational efficiency and high scalability. We present the formulation of the NPD framework, the partitioning strategy for the domain, and the detailed implementation scheme. In addition, we provide theoretical insights into the convergence properties of NPD estimators.

3.1. DDC-based Distributed Learning

The proposed model-fitting approach relies on a trivariate spline over triangulation, where the triangulation serves as a partition of the entire domain. This characteristic naturally prompts us to consider the DDC technique in our distributed computing algorithm. In this section, we develop a distributed learning structure based on DDC techniques, which enables computations to be performed independently on each subregion of the domain. In the following, we provide a detailed description of the DDC-based NPD learning framework.

In contrast to solving a large system of equations as in the global method, our NPD learning algorithm takes a distinct approach by decomposing the problem into a set of subproblems. This decomposition involves partitioning the domain Ω into disjoint connected sets, denoted as $\{\omega_j^0\}_{j=1}^\kappa$. Recognizing that observational units in close proximity often display higher similarity, we introduce additional layers of tetrahedra as a boundary buffer to account for the influence from neighboring subregions. Simultaneously, these buffer zones serve to mitigate potential boundary issues within the subproblems. Specifically,



(a) ω^0 (b) $\text{NBOR}^1(\omega^0)$ (c) $\text{NBOR}^2(\omega^0)$

Figure 1. Illustration of a subregion and its various layers of neighbors.

given a triangulation $\triangle$, we recursively define the ιth layer neighbor (NBOR) of a subregion ω_j^0 as the collection of tetrahedra $T \in \triangle$ such that T shares at least one vertice or a common edge with the $(\iota-1)$th layer neighbor of ω_j^0, that is $\omega_j^\iota = \text{NBOR}^\iota(\omega_j^0) \equiv \cup\{T \in \triangle : T \cap \text{NBOR}^{\iota-1}(\omega_j^0) \neq \emptyset\}$. An example of a subregion of a horseshoe domain, along with its various layers of neighbors, is illustrated in Figure 1. For convenience, we simplify the notation by dropping the subscript ι and denote ω_j as the union of the subregion ω_j^0 and its surrounding tetrahedra. Local estimators related to the subregion are obtained based on information within the subregion ω_j.

For each subregion ω_j, we define $\mathcal{S}_d^r(\triangle_j)$ as the trivariate spline space on triangulation $\triangle_j$, represented by $\mathcal{S}_d^r(\triangle_j) = \{s \in \mathcal{C}^r(\omega_j) : s|_T \in \mathcal{P}_d(T), T \in \triangle_j\}$. Let $\mathcal{M}_j$ denote the index set of basis functions in $\mathcal{S}_d^r(\triangle_j)$. We construct the matrix of basis functions $\mathbf{B}_j(\mathbf{X}_i) = \{B_\xi(\mathbf{X}_i), \xi \in \mathcal{M}_j\}^\top$ for each observation $\mathbf{X}_i$ in the subregion ω_j. Similarly to (3), we seek to find the minimizer $\widehat{\boldsymbol{\gamma}}_j$ of the following optimization problem within each subregion:

$$\min \sum_{\mathbf{X}_i \in \omega_j} \left\{ Y_i - \mathbf{B}_j(\mathbf{X}_i)^\top \boldsymbol{\gamma}_j \right\}^2 + \lambda_j \boldsymbol{\gamma}_j^\top \mathbf{P}_j \boldsymbol{\gamma}_j \tag{5}$$

subject to $\mathbf{H}_j \boldsymbol{\gamma}_j = \mathbf{0}$. Here, $\mathbf{H}_j$ encompasses the smoothness conditions for the spline space $\mathcal{S}_d^r(\triangle_j)$, and $\mathbf{P}_j$ is a block diagonal penalty matrix that satisfies $\boldsymbol{\gamma}_j^\top \mathbf{P}_j \boldsymbol{\gamma}_j = \mathcal{E}(\mathbf{B}_j^\top \boldsymbol{\gamma}_j)$.

We denote the TPST estimator based on the data within subregion ω_j as $\widehat{m}_j(\boldsymbol{x}) = \mathbf{B}_j(\boldsymbol{x})^\top \widehat{\boldsymbol{\gamma}}_j$. In particular, the matrices $\mathbf{B}_j(\boldsymbol{x})$, $\mathbf{P}_j$, and $\mathbf{H}_j$ are submatrices of $\mathbf{B}(\boldsymbol{x})$, $\mathbf{P}$, and $\mathbf{H}$, respectively, with the number of columns proportional to the number of tetrahedra within $\triangle_j$. Consequently, the NPD algorithm significantly reduces the number of observations, unknown parameters to be estimated, and memory requirements compared to considering the complete set of observations in (5).

To aggregate the TPST estimators obtained from the individual subregions, we extract the coefficients of the basis functions defined on ω_j^0, denoted as $\widehat{\boldsymbol{\gamma}}_j$. These coefficients are then stacked together to form a coefficient estimator $\widehat{m}^{\text{D}*}(\boldsymbol{x}) = \mathbf{B}(\boldsymbol{x})^\top \widehat{\boldsymbol{\gamma}}^{\text{D}*}$, where $\mathbf{B}(\boldsymbol{x})$ represents the complete basis function matrix.

Recognizing that each of these estimators corresponds to a polynomial function defined on each triangle, it is essential to address the potential discontinuities at the edges of the tetrahedra. To obtain a smooth estimator for m, that is both close to $\widehat{m}^{\text{D}*}(\boldsymbol{x})$ and belongs to the spline space $\mathcal{S}_d^r(\triangle)$, we leverage the unique characteristic of the TPST method, which achieves smoothness by imposing linear constraints on the spline coefficients.

Inputs: Domain Ω, degree d, smoothness parameter r, and observed dataset $\{(Y_i, \mathbf{X}_i)\}_{i=1}^n$.

Initialize: Construct a triangulation partition $\triangle$ of the domain Ω.

Outputs: Estimated coefficients $\widehat{\boldsymbol{\gamma}}_j^{\text{D}}$.

Step 1. The master processor allocates sub-problems of subregions to different worker processors with a balanced workload.

Step 2. Worker processors execute in parallel. For the subproblem on subregion ω_j, the worker identifies its ℓ layer neighborhood within triangulation $\triangle$ and collects the data within ω_j^ℓ. Based on the data within the subregion, the worker processor performs the TPST smoothing method and sends the model-fitting results to the master processor.

Step 3. The master processor follows the procedures in Section 3.1 to aggregate local estimators and make smoothness adjustments to yield the adjusted estimator.

Algorithm 1: The NPD learning algorithm for the TPST estimation.

To achieve a smooth domain decomposition-based estimator, we aim to find the coefficient vector closest to $\widehat{\boldsymbol{\gamma}}^{\text{D}*}$ within the linear space $\mathcal{H} = \{\boldsymbol{\gamma} : \mathbf{H}\boldsymbol{\gamma} = \mathbf{0}\}$. This can be achieved through QR decomposition: $\mathbf{H}^\top = \mathbf{QR} = (\mathbf{Q}_1\ \mathbf{Q}_2)\binom{\mathbf{R}_1}{\mathbf{R}_2}$. Here, the submatrix $\mathbf{Q}_1$ comprises the first ζ columns of $\mathbf{Q}$, where ζ represents the rank of the matrix $\mathbf{H}$. The subspace $\mathcal{H}$ is a space expanded by the vectors in $\mathbf{Q}_2 = \{\mathbf{Q}_{2(1)}, \ldots, \mathbf{Q}_{2(|\mathcal{M}|-\zeta)}\}$. Therefore, the vector closest to $\widehat{\boldsymbol{\gamma}}^{\text{D}*}$ in $\mathcal{H}$ is the projection of $\widehat{\boldsymbol{\gamma}}^{\text{D}*}$ and can be written as $\widehat{\boldsymbol{\gamma}}^{\text{D}} = \mathbf{Q}_2(\mathbf{Q}_2^\top \mathbf{Q}_2)^{-1}\mathbf{Q}_2^\top \widehat{\boldsymbol{\gamma}}^{\text{D}*}$. The final TPST estimator obtained using the proposed DDC algorithm is $\widehat{m}^{\text{D}}(\boldsymbol{x}) = \mathbf{B}(\boldsymbol{x})^\top \widehat{\boldsymbol{\gamma}}^{\text{D}}$, which is referred to as the NPD estimator.

Algorithm 1 outlines the proposed DDC parallel computing procedure as illustrated in Figure 2. By using DDC techniques, the algorithm can be efficiently implemented on distributed-memory or shared-memory systems capable of running distributed-memory programs and provide a flexible solution that can adapt to the unique characteristics of point cloud data. The algorithm's inherent capability to define neighborhoods and seamlessly align with the TPST smoothing makes it particularly

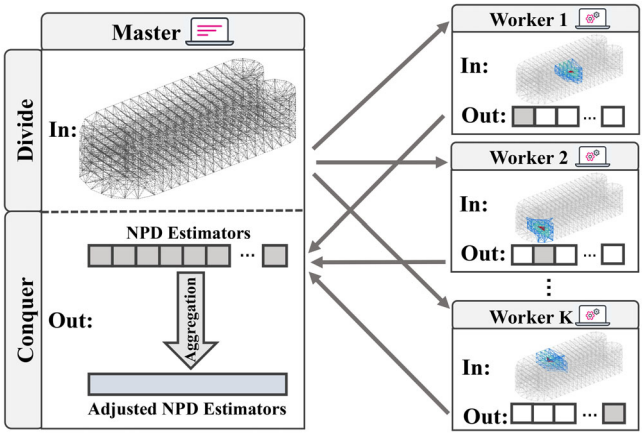


Figure 2. Overview of the NPD learning procedure.

advantageous for processing and analyzing data observed within a point cloud.

In this article, our numerical studies adopt the triangulation partition as the most intuitive and straightforward decomposition strategy, where each tetrahedron serves as an individual subregion, resulting in a total of N subregions.

3.2. Asymptotic Properties of the NPD Estimators

In this section, we analyze the asymptotic properties of the estimators obtained using the domain decomposition method. Specifically, we demonstrate that the NPD-based TPST estimator, denoted as $\widehat{m}^{\mathrm{D}}$, achieves the same convergence rate as the global TPST estimator. To facilitate the analysis, we introduce relevant notations and definitions.

Let $|\triangle_j|$ be the longest edge of the tetrahedron within partition $\triangle_j$ of ω_j, and it holds that $|\triangle_j| \le |\triangle|$, for all $j = 1, \ldots, N$. In the following, we use the notation $\mathcal{D}$ to represent either the entire 3D domain of the data, Ω, or a specific subregion $\omega \subset \Omega$. Let $A(\mathcal{D})$ be the volume of $\mathcal{D}$ and $n(\mathcal{D})$ be the number of points observed within the region $\mathcal{D}$. For simplicity of notation, we denote $n_j = n(\omega_j)$. In addition, we define the Sobolev space $\mathcal{W}^{d,q}(\mathcal{D}) = \{g : |g|_{k,q,\mathcal{D}} < \infty,\ 0 \le k \le d\}$ for $1 \le q \le \infty$, $d \ge 1$. Furthermore, we define the empirical inner product and norm as $\langle g_1, g_2\rangle_{n,\mathcal{D}} = \{n(\mathcal{D})\}^{-1} \sum_{\mathbf{X}_i \in \mathcal{D}} g_1(\mathbf{X}_i) g_2(\mathbf{X}_i)$ and $\|g_1\|_{n,\mathcal{D}}^2 = \langle g_1, g_1\rangle_{n,\mathcal{D}}$, respectively, for measurable functions g_1 and g_2 on $\mathcal{D}$. The theoretical L_2 inner product and the induced norm are given by $\langle g_1, g_2\rangle_{2,\mathcal{D}} = \mathrm{E}\{g_1(\boldsymbol{X}|\mathcal{D}) g_2(\boldsymbol{X}|\mathcal{D})\} = \int_{\mathcal{D}} g_1(\boldsymbol{x}) g_2(\boldsymbol{x}) f_{\boldsymbol{x}|\mathcal{D}}(\boldsymbol{x}) d\boldsymbol{x}$ and $\|g_1\|_{2,\mathcal{D}}^2 = \langle g_1, g_1\rangle_{2,\mathcal{D}} = \int_{\mathcal{D}} g_1^2(\boldsymbol{x}) f_{\boldsymbol{x}|\mathcal{D}}(\boldsymbol{x}) d\boldsymbol{x}$, where $\boldsymbol{X}|\mathcal{D}$ is the random variable restricted on domain $\mathcal{D}$ with density function $f_{\boldsymbol{x}|\mathcal{D}}(\boldsymbol{x}) = \{\int_{\mathcal{D}} f_{\boldsymbol{x}}(\boldsymbol{x}') d\boldsymbol{x}'\}^{-1} f_{\boldsymbol{x}}(\boldsymbol{x})$. In general, for any function g over the closure of domain $\mathcal{D}$, we denote the L_q norm ($1 \le q < \infty$) and the supremum norm as $\|g\|_{L_q(\mathcal{D})} = \left\{\int_{\mathcal{D}} |g(\boldsymbol{x})|^q d\boldsymbol{x}\right\}^{1/q}$ and $\|g\|_{\infty,\mathcal{D}} = \sup_{\boldsymbol{x} \in \mathcal{D}} |g(\boldsymbol{x})|$, respectively. Next, for any $k \ge 0$, denote $|g|_{k,q,\mathcal{D}} = \left\{\sum_{|\boldsymbol{\alpha}|=k} \|D^{\boldsymbol{\alpha}} g\|_{L_q(\mathcal{D})}^q\right\}^{1/q}$ and $|g|_{k,\infty,\mathcal{D}} = \max_{|\boldsymbol{\alpha}|=k} \|D^{\boldsymbol{\alpha}} g\|_{\infty,\mathcal{D}}$.

In the following, we state the technical assumptions used in Theorems 1 and 2.

(A1) The density function of $\mathbf{X}$, $f_{\mathbf{X}}(\cdot)$, is bounded away from zero and infinity.

(A2) The error term $\{\varepsilon_i\}_{i=1}^n$ satisfy $\mathrm{E}\{\varepsilon_i | \mathbf{X}_i = \boldsymbol{x}\} = 0$ and for some $\varsigma \in (1/2, \infty)$, $\sup_{\boldsymbol{x}} \mathrm{E}\{|\varepsilon_i|^{2+\varsigma} | (\mathbf{X}_i = \boldsymbol{x})\} < \infty$. The standard deviation function $\sigma(\boldsymbol{x})$ is continuous on Ω and $0 < c_\sigma \le \inf_{\boldsymbol{x} \in \Omega} \sigma(\boldsymbol{x}) \le \sup_{\boldsymbol{x} \in \Omega} \sigma(\boldsymbol{x}) \le C_\sigma < \infty$.

(A3) The trivariate function $m \in \mathcal{W}^{d+1,\infty}(\Omega)$.

(A4) The triangulation $\triangle$ is π-quasi-uniform, that is, there exists a positive constant π such that the triangulation $\triangle$ satisfies for all $T \in \triangle$, $|\triangle|/\rho_T \le \pi$, where ρ_T is the inscribed circle radius of tetrahedron T.

(A5) The number of the tetrahedra N and the sample size n satisfy that $N = Cn^\eta$ for some constant $C > 0$ and $\eta < \varsigma/(2+\varsigma)$, where ς is given in the Assumption (A2).

(A6) For each subregion ω_j, the triangulation size $|\triangle_j|$ and the volume of the subregion $A(\omega_j)$ satisfy the following: $|\triangle_j| \to 0$ and $n_j^{-1}(\log n_j) A(\omega_j) |\triangle_j|^{-3} \to 0$.

(A7) The roughness tuning parameter λ satisfies $\lambda n^{-1} (\log n)^{-1/2} |\triangle|^{-11/2} \to 0$. For each subregion ω_j, λ_j satisfy that $\lambda_j n_j^{-1/2} (\log n_j)^{-1/2} A^{1/2}(\omega_j) |\triangle_j|^{-11/2} \to 0$.

The above assumptions are mild conditions that can be satisfied in many practical situations. Assumptions (A1)–(A4) are standard in nonparametric literature (Lai and Wang 2013; Yu et al. 2020). Assumption (A1) ensures that the observations are scattered randomly throughout the domain. Assumption (A4) requires that the tetrahedra be relatively uniform in that all angles are similar within the triangulation $\triangle$. Assumptions (A1) and (A4) guarantee that each tetrahedron contains a similar number of observations, and thus leads to a balanced workload for each worker in parallel computing. In addition, sec. 3 of Lai and Schumaker (2007) demonstrated that for $d \ge 8r+1$, the spline space possesses a stable basis, and Lai (1989) established $d \ge 6r+3$ as an alternative sufficient condition. When $d < 8r+1$ or $d < 6r+3$, basis stability can still be achieved if the tetrahedral partition exhibits specific properties. For instance, a partition that is mostly equally spaced can ensure stability. Therefore, in this article, we require that the triangulation used in the proposed method be π-quasi-uniform, as specified in Assumption (A4). Assumption (A5) discusses the requirement of the number of tetrahedra within the triangulation for the domain. Given that the domain area Ω is fixed, we can interchange the number of tetrahedra, N, and the triangulation size, $|\triangle|^{-3}$, in our theoretical results. Assumption (A6) further discusses the requirement of the triangulation size for each subregion. Meanwhile, Assumption (A7) requires a small λ_j for a negligible bias induced by the roughness penalty.

Based on the above assumptions, we first establish the L_2 convergence rate of the mean function estimator $\widehat{m}^{\mathrm{D}*}$ constructed using the NPD method before smoothness adjustment. It is important to note that this estimator is a triangle-wise polynomial function without any smoothness constraints. See the supplementary material for detailed proof.

Theorem 1. Suppose that Assumptions (A1)–(A7) hold, then

$$\|\widehat{m}^{\mathrm{D}*} - m\|^2_{L_2(\Omega)} = O_P\Bigg[\sum_{j=1}^{N}\Big\{A(\omega_j)|\triangle_j|^{2d+2}|m_j|^2_{d+1,\infty,\omega_j}\Big\} + \sum_{j=1}^{N}\Big\{n_j^{-1}A^2(\omega_j)|\triangle_j|^{-3}\Big\}\Bigg], \tag{6}$$

$$\|\widehat{m}^{\mathrm{D}*} - m\|_{\infty,\Omega} = O_P\Big\{|\triangle|^{d+1}|m|_{d+1,\infty,\Omega} + n^{-1/2}(\log n)^{1/2}|\triangle|^{-3/2}\Big\}. \tag{7}$$

Corollary 1. By the definition of $|\triangle_j|$ and $|\triangle|$, we have $|\triangle| \geq |\triangle_j|$. According to Assumption (A4), $|\triangle|/\rho_T \leq \pi$ for all $T \in \triangle$. As ρ_T is the radius of the inscribed circle of the tetrahedron T and is smaller than the longest edge of T, for any $j = 1, \ldots, N$, there exists ρ_T such that $\rho_T \leq |\triangle_j|$ and $|\triangle|/|\triangle_j| \leq |\triangle|/\rho_T \leq \pi$ hold. Therefore, $|\triangle|$ and $|\triangle_j|$ are in the same order. In addition, according to Assumption (A1), the number of points with a subregion ω_j is proportional to the volume of ω_j. Then, we have

$$\|\widehat{m}^{\mathrm{D}*} - m\|^2_{L_2(\Omega)} = O_P\left(|\triangle|^{2d+2}|m|^2_{d+1,\infty,\Omega} + n^{-1}|\triangle|^{-3}\right).$$

Theorem 2 states the L_2 convergence rate of the nonparametric distributed estimator with smoothness adjustment $\widehat{m}^{\mathrm{D}}(\cdot)$. See the supplementary material for detailed proof.

Theorem 2. Suppose that Assumptions (A1)–(A7) hold, then,

$$\|\widehat{m}^{\mathrm{D}} - m\|^2_{L_2(\Omega)} = O_P\left(|\triangle|^{2d+2}|m|^2_{d+1,\infty,\Omega} + n^{-1}|\triangle|^{-3}\right), \tag{8}$$

$$\|\widehat{m}^{\mathrm{D}} - m\|_{\infty,\Omega} = O_P\Big\{|\triangle|^{d+1}|m|_{d+1,\infty,\Omega} + n^{-1/2}(\log n)^{1/2}|\triangle|^{-3/2}\Big\}. \tag{9}$$

Remark 1. Theorem 2 demonstrates that the distributed estimator achieves a convergence rate comparable to that of the global estimator, using multivariate spline over triangulation for both methods. Li et al. (2024) have shown that, under Assumptions (A1)–(A5), if $|\triangle| \asymp n^{-1/(2d+5)}$, the global TPST estimator, $\widehat{m}^{\mathrm{G}}$, achieves the optimal convergence rate as shown in Stone (1982) in the nonparametric regression framework. Given that the distributed TPST estimator $\widehat{m}^{\mathrm{D}}$ is constructed under the same spline smoothing over triangulation framework as $\widehat{m}^{\mathrm{G}}$, it inherits these properties, thereby also achieving the optimal convergence rate under Assumptions (A1)–(A7).

3.3. Implementation Details

3.3.1. Selection of Penalty Parameter

In both global and distributed TPST methods, the penalty parameter is crucial to balance bias and variance. A small penalty reduces bias but potentially increases variance, while a large penalty enforces smoother fits, reducing variance but possibly increasing bias. Popular criteria for the selection of penalty parameters, such as generalized cross-validation (GCV) and block cross-validation (BCV) as advocated by Li et al. (2024), usually produce similar results in our simulation experiments.

Unlike the global method, which selects a single penalty parameter for a balanced tradeoff across the entire domain, the NPD framework assigns distinct penalty parameters to each subregion based on local observations. This approach provides TPST with the flexibility to regulate local smoothness, allowing customized estimations based on varying smoothness requirements across regions.

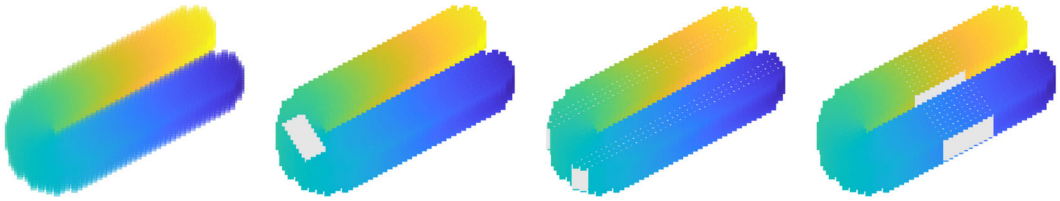
3.3.2. Selection of Triangulation and Subregions

Triangulation is pivotal for accurately approximating complex 3D domains. The required level of triangulation needed is affected by the complexity of the domain and the characteristics of the functions. Highly curved or intricate domains require finer triangulation for accurate representation, potentially increasing computational cost. Equally critical is the requirement for the chosen triangulation to effectively capture the inherent pattern of the function. Functions with rapid changes may also necessitate finer triangulation; function-dependent triangulation often requires knowledge of the signal at multiple locations, which may pose limitations.

As indicated by the assumptions in theoretical development, the choice of triangulation is also influenced by the sample size, with larger samples enabling finer triangulation for more precise estimates. Although the global method benefits from larger samples, the distributed method may face challenges due to data sparsity in point clouds. Consequently, subregion selection becomes necessary for the distributed method, supplemented by incorporating buffer zones designed to accommodate local regions characterized by data sparsity. In practice, we recommend users begin with two layers of neighboring tetrahedra and increase the layer count where needed.

Although constructing a suitable triangulation is crucial for the proposed TPST methods to balance accuracy and computational efficiency, practitioners have some flexibility without compromising estimation performance. The numerical studies in Section 4.1 indicate that both global and distributed estimators are minimally affected by the triangulation choice as long as it is sufficiently fine to capture the underlying pattern. Therefore, practitioners can choose a triangulation level that accurately captures the underlying pattern without incurring an undue computational burden.

In addition to triangulation, subregion selection is essential to ensure efficient processing by individual workers. Subregions can be single tetrahedra or spatially contiguous groupings chosen based on factors such as computational resources and data complexity. In our numerical studies, each tetrahedron is assigned as an autonomous subregion. This approach, in general, is intuitive and well-suited for our numerical studies; however, alternative methods of subregion selection may be considered in different contexts to optimize computational efficiency and resource utilization. In practical scenarios, a more advanced approach involves concurrently selecting triangulation and subregions guided by established criteria such as GCV or BCV.



(a) no missing (b) 5.3% missing (c) 12.0% missing (d) 21.7% missing

Figure 3. True function $m(\cdot)$ with different level of sparsity.

3.3.3. Selection of Smoothing Parameters

Another crucial determinant for achieving accurate estimates is the selection of smoothing parameters for the spline space $\mathcal{S}_d^r(\triangle)$. Parameters d and r can be predetermined or left to the user's discretion. The smoothness parameter r controls the smoothness of the estimated functions. In practice, not all underlying functions require high levels of smoothness. Consequently, we set $r = 1$ in our numerical studies.

The degree of spline polynomials (d) influences the estimator flexibility, where higher values may lead to overfitting and lower values to underfitting. The choice of d depends on the intended interpretation of the estimated function. For a simpler model, a lower d may be suitable, while a higher d preserves local structures for accurate regression. The simulation study in Section 4.1 suggests flexibility in choosing d based on the characteristics of the data. In practice, the degree of spline basis d can be selected using GCV or BCV, similar to the process for the smoothing penalty parameter λ.

4. Numerical Studies

In this section, we conduct two numerical studies to assess the performance of the proposed methods, TPST estimation, and NPD learning, to extract signals from 3D point cloud data. The first study uses simulated point cloud data generated within an irregular 3D horseshoe domain, as illustrated in Figure 4. The second study focuses on biomedical imaging data obtained from PET imaging, which represents a specific instance of 3D point data analysis.

4.1. A Simulation Study

In all simulation examples in this section, the data are generated from the following model: $Y_i = m(\mathbf{X}_i)+\varepsilon_i, \mathbf{X}_i \in \Omega, i = 1, \ldots, n$, where Ω is an irregular 3D horseshoe domain. Figure 3(a) displays both the domain and the true regression function $m(\cdot)$. The random noises, ε_i's, are independent, identically distributed and generated from $N(0, \sigma^2)$, where σ is set to 5 or 10. Furthermore, this simulation study explores the influence of sample size $n =$ 50K, 100K, or 500K.

The point cloud data are geographically situated within the confines of a horseshoe-shaped domain. Within this domain, we implement a randomized design in which observations are generated randomly across the entire spatial extent. To comprehensively assess the performance of our proposed method in handling data sparsity in specific regions, we introduce three distinct sparse patterns characterized by missing rates ranging from 5% to 21.7%. These patterns include (i) one missing block from top to bottom, (ii) two horizontal missing blocks across the left and right half of the horseshoe domain, and (iii) a massive block in the middle across the two parts of the horseshoe. See Figure 3(b)–(d) for visual reference of these three missing patterns, respectively.

To evaluate the influence of triangulation and the degree of spline polynomials on the TPST and NPD estimation methods, we examine two specific triangulations of the horseshoe domain, denoted as $\triangle_1$ and $\triangle_2$, visually represented in Figure 4(b) and (c). The triangulation $\triangle_1$ comprises 504 tetrahedrons and 207 vertices, while $\triangle_2$ consists of 756 tetrahedrons and 276 vertices, providing varying levels of mesh granularity. We explore two distinct degrees of spline, specifically $d = 2$ and $d = 3$, in conjunction with the two different triangulations, $\triangle_1$ and $\triangle_2$. To facilitate this comparison, we maintain a consistent value of $r = 1$. The results of this investigation are presented in Table C.1 in the supplementary material, where the observed differences between scenarios $d = 2$ and $d = 3$ appear to be negligible. As such, for brevity and clarity, we opt to present our primary results exclusively with $d = 3$ in the main text.

In addition, we compare the proposed TPST and NPD methods with traditional thin plate spline (TPS1) and tensor product spline (TPS2) methods. Both TPS1 and TPS2 are implemented using the `gam()` function within the R package `mgcv`, with the smoothing parameters chosen appropriately to optimize estimation and prediction accuracy.

To assess the estimation performance of each method, we compute the mean integrated squared error (MISE) for the estimator of $m(\cdot)$. Table 1 provides the average MISEs over 100 replications across various scenarios. First, as the sample size increases or the noise level σ decreases, all methods exhibit enhanced accuracy in estimation. Second, irrespective of the scenarios, both the TPST and NPD methods consistently outperform the traditional estimation methods TPS1 and TPS2. This superiority arises from the adeptness of the proposed TPST and NPD methods in handling intricate domains and regions within 3D point clouds. Third, the TPST and NPD estimators demonstrate similar performance across two partitions, suggesting that finer tetrahedrons do not significantly enhance the estimation when the number of tetrahedra is already sufficient to capture data patterns. Fourthly, in scenarios with low or no missing rates, the TPST and NPD estimators perform similarly, regardless of the sample size or noise level. As the sparse region expands, the distributed learning method may face more challenges, but these difficulties can be mitigated by a larger dataset.

In the scenario with complete data without missing observations, Figure 4(e)–(h) exhibit the estimated functions using

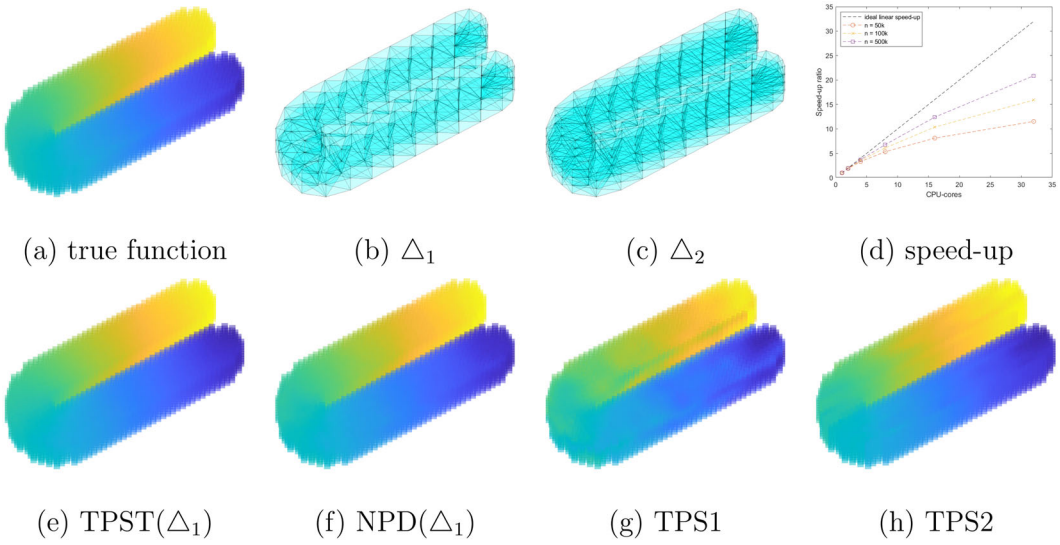


Figure 4. True function and its estimation results based on different methods: (a) true function $m(\cdot)$, (b) coarse triangulation $\triangle_1$, (c) fine triangulation $\triangle_2$, (d) speed-up of the distributed trivariate spline estimator over triangulation $\triangle_1$, (e) global trivariate spline estimator over triangulation $\triangle_1$, (f) distributed trivariate spline estimator over triangulation $\triangle_1$, (g) thin-plate spline estimator, and (h) tensor product spline estimator.

Table 1. Mean integrated squared errors (MISEs) of different methods across different sample sizes (n), noise levels (σ), and levels of sparsity indicated by missing rate (mr).

(n,σ)	mr	NPD		TPST		TPS1	TPS2	(n,σ)	mr	NPD		TPST		TPS1	TPS2
		$\triangle_1$	$\triangle_2$	$\triangle_1$	$\triangle_2$					$\triangle_1$	$\triangle_2$	$\triangle_1$	$\triangle_2$		
(50K, 5)	0.0%	0.037	0.039	0.038	0.036	0.221	0.079	(50K, 10)	0.0%	0.105	0.122	0.093	0.092	0.476	0.158
	5.3%	0.040	0.042	0.039	0.037	0.228	0.085		5.3%	0.115	0.133	0.096	0.095	0.490	0.172
	12.0%	0.040	0.039	0.041	0.038	0.229	0.083		12.0%	0.114	0.133	0.098	0.097	0.504	0.167
	21.7%	0.059	0.061	0.048	0.045	0.324	0.089		21.7%	0.174	0.199	0.111	0.110	0.644	0.176
(100K, 5)	0.0%	0.023	0.024	0.023	0.022	0.148	0.060	(100K, 10)	0.0%	0.063	0.069	0.063	0.062	0.327	0.110
	5.3%	0.025	0.026	0.024	0.023	0.152	0.063		5.3%	0.067	0.075	0.065	0.064	0.335	0.118
	12.0%	0.025	0.026	0.025	0.024	0.152	0.063		12.0%	0.068	0.075	0.066	0.065	0.341	0.117
	21.7%	0.037	0.037	0.029	0.028	0.226	0.068		21.7%	0.099	0.109	0.078	0.076	0.451	0.122
(500K, 5)	0.0%	0.008	0.008	0.007	0.007	0.056	0.029	(500K, 10)	0.0%	0.021	0.021	0.022	0.022	0.128	0.055
	5.3%	0.008	0.009	0.007	0.007	0.057	0.030		5.3%	0.022	0.022	0.023	0.023	0.132	0.058
	12.0%	0.008	0.009	0.007	0.007	0.058	0.031		12.0%	0.022	0.023	0.023	0.023	0.133	0.058
	21.7%	0.014	0.013	0.009	0.009	0.106	0.036		21.7%	0.033	0.032	0.029	0.029	0.198	0.064

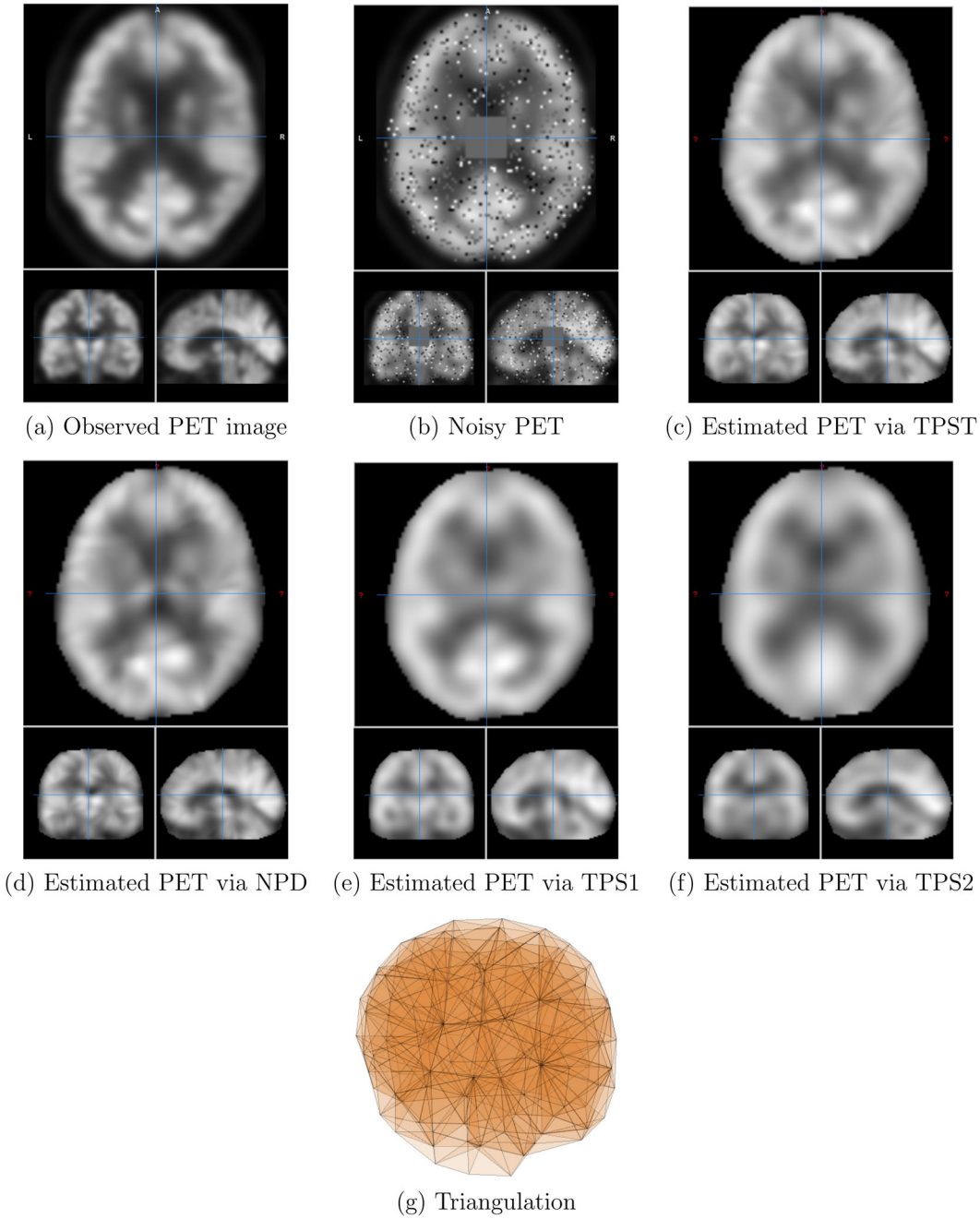
TPST: global trivariate spline over triangulation estimator; NPD: distributed trivariate spline over triangulation estimator; TPS1: thin-plate spline estimator; and TPS2: tensor product spline estimator.

various methods, all based on a sample size of 50K. Specifically, Figure 4(e) and (f) showcase the TPST and NPD estimators based on triangulation $\triangle_1$. It is evident that there are minimal performance differences between the TPST and NPD estimators. Figure 4(g) and (h) depict the TPS1 and TPS2 estimates based on the same simulated data. These plots demonstrate that both the TPST and NPD estimators offer a closer approximation to the true function when compared to the TPS1 and TPS2 methods, which is notably visible as the former exhibit a less blurry effect.

Finally, we assess the efficiency gains achieved by the proposed NPD learning method, a crucial aspect of parallel programming. Speed-up is widely used in parallel programming and is calculated by the serial execution time (CPU-cores = 1) divided by the execution time where # of CPU-cores = # nodes × # tasks-per-node × # cpus-per-task. All experiments are conducted on a high-performance computing system equipped with Intel CPUs. As illustrated in Figure 4(d), when considering sample sizes of n = 50K, 100K, and 500K, the NPD method exhibits speed-up ratios of more than 12.2%, 15.1%, and 21% when transitioning from CPU-core = 1 to CPU-core = 32, respectively. Obviously, the larger the sample size, the more significant the performance of NPD.

4.2. Biomedical Imaging Analysis

Alzheimer's disease (AD) has become the primary cause of dementia today. Advancing neuroimaging techniques is critically important, particularly to significantly enhance the early diagnosis of AD. Among these techniques, PET imaging stands as a cornerstone. A PET image is composed of voxels, each with 3D spatial coordinates and an intensity value representing the concentration of the radioactive tracer. Consequently, PET image data can be efficiently represented as a point cloud, facilitating further analysis such as data compression, visualization, denoising, and image inpainting. However, conventional PET analysis approaches face challenges in handling the high resolution of brain images, highlighting the need for effective image reconstruction methods. In our study, we employ both the TPST and NPD methods, which are vital to denoising,



(a) Observed PET image (b) Noisy PET (c) Estimated PET via TPST

(d) Estimated PET via NPD (e) Estimated PET via TPS1 (f) Estimated PET via TPS2

(g) Triangulation

Figure 5. An illustration of an observed PET image, the PET image with missing values, missing region, and salt-and-pepper noise, estimated PET image using the global method (TPST), distributed method (NPD), thin-plate spline (TPS1) and tensor product spline (TPS2), as well as triangulations of the brain domain.

capturing local features, improving resolution, and performing image inpainting for authentic PET images, all while optimizing storage efficiency.

To demonstrate the effectiveness of our proposed approaches in denoising, capturing local features, elevating resolution, and image inpainting, we randomly select 15 PET images from the ADNI study (*https://adni.loni.usc.edu*) as representative examples. Figure 5(a) illustrates one of such PET images, showing the axial, coronal, and sagittal planes. For each image, we simulate two missing data scenarios: randomly missing voxels and block-wise missing voxels with an overall missing rate of 10%.

The simulation results in Section 4.1 suggest that the choice of triangulation has a minimal effect on the performance of the TPST estimator, provided the triangulation is fine enough to capture the underlying pattern. Therefore, for the brain imaging analysis, we have selected the triangulation shown in Figure 5(g), which comprises 317 tetrahedra and 117 vertices, after exploring several triangulation options. To effectively capture high-frequency oscillations in human brain scans, we set the polynomial degree d to 6.

In addition, we examine the ability to denoise, capture local features, and perform image inpainting using the global (TPST) and distributed (NPD) methods. We further compare them with traditional thin-plate spline (TPS1) and tensor product spline (TPS2) estimations with a similar number of basis functions. Figure 5(c)–(f) present the estimated PET images obtained

Table 2. Estimation accuracy results of global and distributed trivariate spline estimators based on measures: root mean squared error (RMSE), mean absolute error (MAE), mean relative absolute error (MRAE), and peak signal-to-noise ratio (PSNR).

Method	RMSE	MAE	RMAE	PSNR
TPST	0.0643	0.0496	0.0662	72.0015
NPD	0.0639	0.0491	0.0676	72.0490
TPS1	0.0882	0.0693	0.0925	69.2645
TPS2	0.1154	0.0893	0.1168	66.9166

TPST: global trivariate spline over triangulation estimator; NPD: distributed trivariate spline over triangulation estimator; TPS1: thin-plate spline estimator; and TPS2: tensor product spline estimator.

through these four different methods. In particular, both TPST and NPD exhibit superior capabilities compared to traditional TPS1 and TPS2 methods. Moreover, NPD performs better in denoising, retaining essential local features, and inpainting brain structures. This enhanced performance is likely due to its flexibility in choosing local tuning parameters for different subregions, allowing it to better align with the local smoothness characteristics of the true signal.

To test the accuracy of different methods, we treat the observed PET image shown in Figure 5(a) as the underlying truth and denote W_i as the true image value for the ith voxel, $i = 1, \ldots, n$, within the human brain domain. Then, we assess the estimated values, $\widehat{W}_i$'s, derived from a noisy image like the one shown in Figure 5(b) using different methods. For this evaluation, we consider the following four distinct measures: (i) average root mean squared error (RMSE $= \{n^{-1}\sum_{i=1}^{n}(\widehat{W}_i - W_i)^2\}^{1/2}$), (ii) average mean absolute error (MAE $= n^{-1}\sum_{i=1}^{n}|\widehat{W}_i - W_i|$), (iii) average peak signal-to-noise ratio (PSNR $= 20\log 10\{\max(W_i)/\text{RMSE}\}$) and (iv) average mean relative absolute error (MRAE $= n^{-1}\sum_{i=1}^{n}|(\widehat{W}_i - W_i)/W_i|$), calculated across 15 randomly selected PET images. The results of our proposed TPST and NPD methods, along with those of the traditional TPS1 and TPS2 methods, are summarized in Table 2. As both TPS1 and TPS2 can be influenced by noise outside the actual brain, all measurements are calculated based solely on voxels within the brain domain to ensure a fair comparison. Table 2 illustrates the superior performance of TPST and NPD over traditional TPS1 and TPS2 methods regarding the four distinct metrics defined above, which substantiates our findings from Figure 5. Furthermore, TPST and NPD exhibit comparable performance, and NPD demonstrates a slight advantage over TPST.

5. Conclusions and Discussion

This article has tackled the fundamental challenges of extracting valuable information from point cloud data in a computationally efficient manner. By combining innovative statistical methodology, scalable parallel and distributed computing algorithms, and rigorous theoretical foundations, we have developed a comprehensive nonparametric approach for signal extraction. Our proposed method offers numerous advantages over existing approaches, both in practical implementation and theoretical underpinnings.

From a computational perspective, we have devised an efficient and scalable computing algorithm that can be seamlessly implemented across different computing platforms. This algorithm enables us to process large-scale point cloud datasets effectively, reducing computational time and resource requirements. Theoretically, we have conducted in-depth investigations into the properties of the estimators obtained through our NPD algorithm. The results of our analyses demonstrate that the NPD estimators achieve asymptotic properties similar to those of the global TPST estimator. This theoretical validation provides a solid foundation for the efficacy and reliability of our proposed method.

An avenue of research could focus on incorporating additional covariates or features within the distributed learning framework to enhance the accuracy and robustness of signal extraction from point cloud data. Using the distributed nature of the data, it is possible to incorporate information from multiple sources or sensors, allowing for more comprehensive modeling and improved signal extraction performance.

Quantifying uncertainty in the context of point cloud data is an intriguing and intricate challenge. Currently, bootstrapping methods show promise in handling uncertainty for global methods, particularly when dealing with datasets of moderate size. However, these techniques may encounter limitations when applied to larger datasets, where computational demands become a significant concern. One potential area for future exploration lies in harnessing the potential of distributed learning techniques to tackle the computational bottlenecks posed by big data. By developing novel distributed approaches tailored to uncertainty quantification in point cloud data, we can effectively extend our methodology to larger and more complex datasets. This endeavor promises to be a fruitful area of research, offering valuable insights into enhancing the robustness and reliability of point cloud data analysis, especially in the context of massive datasets commonly encountered in modern applications.

Supplementary Materials

Supplementary Material: This supplementary material contains related technical details, proofs of the theoretical results, and additional results in simulation studies. (Supplement.pdf, .pdf)

R-package: R-package contains the code to perform the global and distributed trivariate penalized spline estimation methods described in the article. In addition, it contains the data generator used in the simulation examples in the article, along with a README file providing a detailed description of its contents. (Code.zip, .zip)

Acknowledgments

The authors gratefully acknowledge Dr. Ming-Jun Lai for providing the MATLAB code for spline basis constructions. The authors also thank the editor, Dr. Faming Liang, an associate editor, and two anonymous reviewers for their constructive comments, which substantially improved the article.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Data Availability Statement

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (*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 the 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*

Funding

The authors gratefully acknowledge partial support for this research from the National Institutes of Health under grant 1R01AG085616. Li Wang's research was additionally supported by the National Science Foundation under grants DMS-2203207 and DMS-2426173. Guannan Wang's research also received partial support from the Simons Foundation under grant 963447.

References

Banerjee, M., Durot, C., and Sen, B. (2019), "Divide and Conquer in Non-standard Problems and the Super-Efficiency Phenomenon," *The Annals of Statistics*, 47, 720–757. [747]

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2018), "Distributed Testing and Estimation Under Sparse High Dimensional Models," *The Annals of Statistics*, 46, 1352–1382. [747]

Chen, X., Liu, W., and Zhang, Y. (2022), "First-Order Newton-Type Estimator for Distributed Estimation and Inference," *Journal of the American Statistical Association*, 117, 1858–1874. [747]

Cheng, S. W., Dey, T. K., Ramos, E. A., and Ray, T. (2005), "Quality Meshing of Polyhedra with Small Angles," *International Journal of Computational Geometry & Applications*, 15, 421–461. [748]

Chung, M. K., Wang, Y., and Wu, G. (2018), "Discrete Heat Kernel Smoothing in Irregular Image Domains," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5101–5104. [747]

Fang, Q., and Boas, D. A. (2009), "Tetrahedral Mesh Generation from Volumetric Binary and Grayscale Images," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1142–1145. [748]

Gao, Y., Liu, W., Wang, H., Wang, X., Yan, Y., and Zhang, R. (2022), "A Review of Distributed Statistical Inference," *Statistical Theory and Related Fields*, 6, 89–99. [747]

Goldsmith, J., Huang, L., and Crainiceanu, C. M. (2014), "Smooth Scalar-on-Image Regression via Spatial Bayesian Variable Selection," *Journal of Computational and Graphical Statistics*, 23, 46–64. [747]

Green, P. J., and Silverman, B. W. (1994), "*Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, London: Chapman and Hall. [749]

Guo, J., Duan, X., and Shum, C. (2010), "Non-Isotropic Gaussian Smoothing and Leakage Reduction for Determining Mass Changes Over Land and Ocean Using GRACE Data," *Geophysical Journal International*, 181, 290–302. [747]

Huo, S., Liu, G., Zhang, J., and Song, C. (2020), "A Smoothed Finite Element Method for Octree-based Polyhedral Meshes with Large Number of Hanging Nodes and Irregular Elements," *Computer Methods in Applied Mechanics and Engineering*, 359, 112646. [747]

Jamin, C., Alliez, P., Yvinec, M., and Boissonnat, J.-D. (2015), "CGALmesh: A Generic Framework for Delaunay Mesh Generation," *ACM Transactions on Mathematical Software (TOMS)*, 41, 1–24. [748]

Jiang, S., Cao, J., Colditz, G. A., and Rosner, B. (2021), "Predicting the Onset of Breast Cancer Using Mammogram Imaging Data with Irregular Boundary," *Biostatistics*, 24, 358–371. [747]

Jordan, M. I., Lee, J. D., and Yang, Y. (2019), "Communication-Efficient Distributed Statistical Inference," *Journal of the American Statistical Association*, 114, 668–681. [747]

Kang, Y. (2020), "Consistent Blind Image Deblurring Using Jump-Preserving Extrapolation," *Journal of Computational and Graphical Statistics*, 29, 372–382. [747]

Kim, M., and Wang, L. (2021), "Generalized Spatially Varying Coefficient Models," *Journal of Computational and Graphical Statistics*, 30, 1–10. [747]

Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. (2014), "A Scalable Bootstrap for Massive Data," *Journal of the Royal Statistical Society*, Series B, 76, 795–816. [747]

Lai, M. J. (1989), "On construction of Bivariate and Trivariate Vertex Splines on Arbitrary Mixed Grid Partitions," Ph.D. dissertation, Texas A&M University. [751]

Lai, M. J., and Schumaker, L. L. (2007), *Spline Functions on Triangulations*, Cambridge: Cambridge University Press. [751]

Lai, M. J., and Wang, L. (2013), "Bivariate Penalized Splines for Regression," *Statistica Sinica*, 23, 1399–1417. [749,751]

Li, X., Yu, S., Wang, Y., Wang, G., Wang, L., and Lai, M.-J. (2024), "Non-parametric Regression for 3D Point Cloud Learning," *Journal of Machine Learning Research*, 25, 1–56. [747,752]

Lindgren, F., Rue, H., and Lindström, J. (2011), "An Explicit Link between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach," *Journal of the Royal Statistical Society*, Series B, 73, 423–498. [747]

Meng, C., Xie, R., Mandal, A., Zhang, X., Zhong, W., and Ma, P. (2021), "LowCon: A Design-based Subsampling Approach in a Misspecified Linear Model," *Journal of Computational and Graphical Statistics*, 30, 694–708. [747]

Morris, J. S., and Carroll, R. J. (2006), "Wavelet-based Functional Mixed Models," *Journal of the Royal Statistical Society*, Series B, 68, 179–199. [747]

Persson, P. O., and Strang, G. (2004), "A Simple Mesh Generator in MATLAB," *SIAM Review*, 46, 329–345. [748]

Ramsay, T. (2002), "Spline Smoothing Over Difficult Regions," *Journal of the Royal Statistical Society*, Series B, 64, 307–319. [747]

Reiss, P. T., and Ogden, R. T. (2010), "Functional Generalized Linear Models with Images as Predictors," *Biometrics*, 66, 61–69. [747]

Shang, Z., and Cheng, G. (2017), "Computational Limits of a Distributed Algorithm for Smoothing Spline," *Journal of Machine Learning Research*, 18, 1–37. [747]

Shi, C., Lu, W., and Song, R. (2018), "A Massive Data Framework for M-estimators with Cubic-Rate," *Journal of the American Statistical Association*, 113, 1698–1709. [747]

Si, H. (2015), "TetGen, a Delaunay-based Quality Tetrahedral Mesh Generator," *ACM Transactions on Mathematical Software (TOMS)*, 41, 1–36. [748]

Song, Q., and Liang, F. (2014), "A Split-and-Merge Bayesian Variable Selection Approach for Ultrahigh Dimensional Regression," *Journal of the Royal Statistical Society*, Series B, 77, 947–972. [747]

Stone, J. C. (1982), "Optimal Global Rates of Convergence for Nonparametric Regression," *The Annals of Statistics*, 10, 1040–1053. [748,752]

Szabó, B., and van Zanten, H. (2019), "An Asymptotic Analysis of Distributed Nonparametric Methods," *Journal of Machine Learning Research*, 20, 1–30. [747]

The CGAL Project. (2020). *CGAL User and Reference Manual*. CGAL Editorial Board, 5.1.1 edition. [748]

Volgushev, S., Chao, S.-K., and Cheng, G. (2019), "Distributed Inference for Quantile Regression Processes," *The Annals of Statistics*, 47, 1634–1662. [747]

Wang, C., Chen, M.-H., Schifano, E., Wu, J., and Yan, J. (2016), "Statistical Methods and Computing for Big Data," *Statistics and Its Interface*, 9, 399–414. [747]

Wang, H., and Ranalli, M. G. (2007), "Low-Rank Smoothing Splines on Complicated Domains," *Biometrics*, 63, 209–217. [747]

Wang, H., Zhu, R., and Ma, P. (2018), "Optimal Subsampling for Large Sample Logistic Regression," *Journal of the American Statistical Association*, 113, 829–844. [747]

Wang, L., Wang, G., and Gao, A. S. (2024), "Exploring Heterogeneity and Dynamics of Meteorological Influences on US PM2. 5: A Distributed Learning Approach with Spatiotemporal Varying Coefficient Models," *Spatial Statistics*, 61, 100826. [747]

Wang, S., Cao, G., Shang, Z., and for the Alzheimer's Disease Neuroimaging Initiative. (2021), "Estimation of the Mean Function of Functional Data via Deep Neural Networks," *Stat*, 10, e393. [747]

Wilhelm, M., Dedè, L., Sangalli, L. M., and Wilhelm, P. (2016), "IGS: An IsoGeometric Approach for Smoothing on Surfaces," *Computer Methods in Applied Mechanics and Engineering*, 302, 70–89. [747]

Wong, R. K. W., Lee, T. C. M., Paul, D., Peng, J., and Alzheimer's Disease Neuroimaging Initiative. (2016), "Fiber Direction Estimation, Smoothing and Tracking in Diffusion MRI," *The Annals of Applied Statistics*, 10, 1137–1156. [747]

Wood, S. N. (2003), "Thin Plate Regression Splines," *Journal of the Royal Statistical Society*, Series B, 65, 95–114. [749]

Yu, S., Wang, G., and Wang, L. (2024), "Distributed Heterogeneity Learning for Generalized Partially Linear Models with Spatially Varying Coefficients," *Journal of the American Statistical Association*, 1–15. DOI: 10.1080/01621459.2024.2359131. [747]

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. [751]

Yue, Y., and Speckman, P. L. (2010), "Nonstationary Spatial Gaussian Markov Random Fields," *Journal of Computational and Graphical Statistics*, 19, 96–116. [747]

Zhang, J., Meng, C., Yu, J., Zhang, M., Zhong, W., and Ma, P. (2023), "An Optimal Transport Approach for Selecting a Representative Subsample with Application in Efficient Kernel Density Estimation," *Journal of Computational and Graphical Statistics*, 32, 329–339. [747]

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. [747]