BAYESIAN SPANNING TREE MODELS FOR COMPLEX SPATIAL DATA

A Dissertation

by

ZHAO TANG LUO

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Huiyan Sang
Co-Chair of Committee,	Bani Mallick
Committee Members,	Krishna Narayanan
	Debdeep Pati
Head of Department,	Brani Vidakovic

May 2022

Major Subject: Statistics

ABSTRACT

In many applications, spatial data often display heterogeneous dependence patterns and may be subject to irregular geographic constraints. In light of these challenges, this dissertation develops several novel Bayesian methodologies for modeling non-trivial spatial data.

The first part of this dissertation develops a Bayesian partition prior model for a finite number of spatial locations using random spanning trees (RSTs) of a spatial graph, which guarantees contiguity in clustering and allows to detect clusters with arbitrary shapes and sizes. We embed this model within a hierarchical modeling framework to estimate spatially clustered coefficients and their uncertainty measures in a regression model. We prove posterior concentration results and design an efficient Markov chain Monte Carlo algorithm.

In the second part, we propose a new class of locally stationary stochastic processes, where local spatially contiguous partitions are modeled by a soft partition process via predictive RSTs for flexible cluster shapes. This valid nonstationary process model allows to knit together local models such that both parameter estimation and prediction can be performed under a coherent framework, and to capture both abrupt changes and smoothness in a spatial random field. We study the posterior concentration theories for this Bayesian process model.

Finally, we consider Bayesian ensemble models for nonparametric regression on complex constrained domains. We first propose a Bayesian additive regression model using RST manifold partition models as weak learners, which are capable of capturing any irregularly shaped spatially contiguous partitions while respecting intrinsic geometries and domain boundary constraints. For applications that also involve possibly high dimensional features without known multivariate structures, we further develop a Bayesian additive multivariate decision trees model that combines univariate split rules and novel multivariate split rules in each weak learner. The proposed multivariate split rules are built upon predictive spanning tree bipartition models on reference knots, which are capable of achieving flexible nonlinear decision boundaries on manifold feature spaces while reducing computations.

DEDICATION

To my parents, and to the memory of my grandfather.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my committee chair Dr. Huiyan Sang, not only for her valuable guidance and persistent encouragement, but also for the various opportunities she has provided. Similarly, deep gratitude goes to my co-chair Dr. Bani Mallick, for his remarkable insights and continued support. It was their expertise and enthusiasm that helped me overcome difficult obstacles and grow into a better researcher. I am also grateful to my committee members Drs. Krishna Narayanan and Debdeep Pati, for their insightful comments and suggestions.

I would also like to thank Dr. Nilabja Guha, Changwoo Lee, Se Yoon Lee, and Yan Zhong, for the helpful discussion on my research projects. I am thankful to the computational resource and technical support provided by the Department of Statistics and Texas A&M High Performance Research Computing. Special thanks to my friends and colleagues who accompanied and supported my Ph.D. journey.

Last but not least, I would like to acknowledge with gratitude, the support and encouragement from my parents and fiancée. None of these would be possible without your endless love.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Drs. Huiyan Sang, Bani Mallick and Debdeep Pati of the Department of Statistics and Dr. Krishna Narayanan of the Department of Electrical & Computer Engineering.

The data analyzed for Chapter 2 was provided by Dr. Furong Li of the Ocean University of China.

All work conducted for the dissertation was completed by the student, in collaboration with Dr. Huiyan Sang and Dr. Bani Mallick. The proof of Proposition 2.5 was coauthored with Dr. Huiyan Sang.

Funding Sources

Graduate study was partially supported by National Science Foundation DMS-1737885, DMS-1854655 and CCF-1934904, and a Graduate Assistantship from the Department of Statistics at Texas A&M University.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Spatial data arise from diverse disciplines such as geoscience, meteorology, and environmental science. The main objective of spatial data analysis is to model dependence among observations to facilitate parameter inference and out-of-sample prediction. In many applications, the spatial dependence structure can be fairly complicated. To name a few, the temperature-salinity relationship in ocean water and the precipitation over the contiguous United States can exhibit substantially different spatial patterns across some borders. Accounting for complex spatial dependence can be more challenging for data collected from irregularly shaped domains due to geographic constraints such as lakes and coasts. New methodologies in spatial statistics need to be developed for data with non-trivial spatial patterns.

Spanning trees have recently been demonstrated as an effective modeling tool for spatially varying dependence (Li and Sang, 2019; Teixeira et al., 2015, 2019), as they naturally induce contiguous partitions on a finite set with flexible shapes. Following a similar path, we propose several novel Bayesian models with sound theoretical guarantee and efficient computation algorithms for various non-trivial spatial analysis problems.

In Chapter 2, we consider statistical models where the *latent* variables of interest are assumed to have spatially clustered patterns. One prominent example is the spatially varying coefficient regressions where the coefficients are homogeneous within clusters but can change abruptly across clusters. We propose to use a Bayesian random spanning tree (RST) partition model to guarantee spatial contiguity in clustering, to allow for flexible cluster shapes, and to deliver uncertainty quantification, while most existing methods cannot achieve these at the same time. Bayesian posterior concentration theory and an efficient Markov chain Monte Carlo algorithm are developed for the proposed model.

Nonstationary spatial process models are important tools in spatial statistics as they provide a coherent framework for modeling heterogeneous spatial dependence and out-of-sample prediction. In particular, locally stationary process models are able to adapt to local and nonstationary data features by partitioning the spatial domain into some local subregions. However, there are several challenging questions surrounding them on how to obtain flexible partitions and how to perform prediction near partition boundaries. In Chapter 3, we extend the RST partition model on a finite set to a *soft* partition process model on a spatial domain, upon which we develop a new class of locally stationary Gaussian process models to capture both abrupt changes and smoothness in a spatial random field. We also study the posterior concentration theory concerning the asymptotic behavior of this Bayesian nonstationary process model.

In Chapter 4, we consider a nonparametric regression problem with covariates lying on a complex constrained spatial domain, or more generally, a compact Riemannian manifold. Most existing literature either assumes a globally smooth true function or ignores intrinsic geometries of the domain. We develop a novel ensemble learning method adapting to different local smoothness levels. The proposed model utilizes RST-based manifold partition models as weak learners, which are capable of capturing any irregularly shaped spatially contiguous partitions while respecting intrinsic geometries and domain boundary constraints.

The RST ensemble model in Chapter 4 only considers structured features with *known* multivariate structures (e.g., spatial locations possibly lying on a Riemannian manifold). In many applications such as housing price prediction, it is of interest to also incorporate unstructured features, i.e., features *without* multivariate structures or with *unknown* multivariate structures (e.g., square footage and housing age). In Chapter 5, we develop a new class of Bayesian additive multivariate decision trees models that combine univariate split rules for handling possibly high dimensional unstructured features and novel multivariate split rules for structured features in each weak learner. The proposed multivariate split rules are built upon predictive spanning tree bipartition models on reference knots, which allow for highly flexible nonlinear decision boundaries on manifold feature spaces.

2. A BAYESIAN CONTIGUOUS PARTITIONING METHOD FOR LEARNING CLUSTERED LATENT VARIABLES [*]

2.1 Introduction

Spanning trees have gained popularity as a flexible computing tool in computational geometry (Preparata and Shamos, 2012) and clustering analysis (Zahn, 1970; Grygorash et al., 2006), since they are capable of guaranteeing contiguous clustering configurations and detecting clusters with irregular shapes. A spanning tree of a connected graph is a subgraph connecting all vertices in the graph without cycles, in which any two vertices are connected by exactly one edge. A partition of vertices is induced when some edges in a spanning tree are removed such that vertices connected to each other form a cluster. A large body of existing literature on spanning trees is based on machine learning algorithms directly using observed points or point-level features (e.g., Assunção et al., 2006; Guo, 2008; Aydin et al., 2018), whereas the development of spanning tree based modeling and inference framework involving clustered latent variables is still at its infancy.

Our main contribution is to propose a Bayesian model-based spanning tree partitioning method, along with theoretical justifications and efficient computational algorithms, to model clustered latent variables with a focus on spatially clustered varying coefficient models. Most existing literature in spatial regression assume regression coefficients are constants or smoothly varying in space (Fotheringham et al., 2003; Gelfand et al., 2003; Mu et al., 2018). But in many applications, relationships among spatial variables may change abruptly across some boundaries. There is a great need to detect spatially clustered patterns with uncertainty measures in such relationships that allow practitioners to conduct and interpret subregional analysis. The work in this chapter is among the first to develop a Bayesian approach for detecting contiguous clusters in regression coefficients.

[*] Reprinted from Luo, Z. T., Sang, H., and Mallick, B. (2021b). A Bayesian contiguous partitioning method for learning clustered latent variables. *Journal of Machine Learning Research*, 22(37):1–52. The authors hold the copyright.

The Bayesian Spatially Clustered Coefficient Model (BSCC) uses different spanning trees for each covariate and treats them as unknown parameters. Model specifications of space partitions are done by assigning priors on spanning trees, and then the number and the positions of removed edges given a spanning tree. As a result, it allows an adaptive spatial order for cluster detection. Indeed, we show that the sample space of partitions induced from the Bayesian random spanning tree models accommodates all possible contiguous partitions with arbitrary shapes and sizes, defined from connected components of any given graph. Most existing clustering methods which we will review in Section 2 do not possess this property. We emphasize that this property has two important implications. First, it allows us to simplify a complex combinatorial graph partitioning problem into a more compact tree based prior representation that can facilitate computation while maintaining flexibility. Second, the method enjoys great flexibility in the cluster shapes and naturally induces spatially contiguous clusters so that practitioners can interpret clusters as subregions. And the number of clusters is treated as random and determined from data.

An additional advantage of the BSCC is that the Bayesian inference allows us to assess uncertainties in the position of spatial boundaries and the estimated regression models within clusters. Moreover, although we concentrate on the Gaussian spatial regression models in this chapter, the proposed partitioning prior model is generic and we propose extensions of the method for embedding in and adaption to various Bayesian hierarchical modeling frameworks that involve latent piecewise constant variables. Finally, since the method is built upon graphs such as triangular meshes, it can be used as a flexible prior on non-exchangeable partitions of data or latent variables distributed on graphs/networks in complex geometric domains.

The regression problem we consider in this chapter is high-dimensional in nature with n samples and np unknown regression coefficients. We prove that the proposed model achieves posterior consistency, under an asymptotic framework for piecewise constant functions defined on random graphs with a diverging number of vertices. Theoretical guarantee of

Bayesian binary treed methods is developed recently (Linero and Yang, 2018; Ročková and van der Pas, 2020; Ročková and Saha, 2019). However, to the best of our knowledge, theoretical properties of spanning tree based Bayesian partition models haven't been investigated in the literature.

The inference of the proposed method is performed in a Bayesian framework, where we extend the conventional reversible jump Markov chain Monte Carlo (RJ-MCMC) algorithm (Green, 1995) by employing various computation strategies such as parallel tempering, low-rank matrix operations, Cholesky factor updates/downdates, and collapsed Gibbs sampling that greatly improves the computation efficiency for large data sets. The RJ-MCMC procedure allows partitions and spanning trees to be updated adaptively so it can achieve high accuracy in cluster recovery and coefficient estimation, as evidenced by our numerical results that demonstrate striking improvements over competing methods.

The rest of the chapter is organized as follows. In Section 2.2, we review other related model-based clustering approaches. In Section 2.3, we present the Bayesian Spatially Clustered Coefficient regression model, state the theoretical results, develop computation algorithms for Bayesian model implementation, and discuss hyperparameter selection. In Section 2.4, we present extensions to other hierarchical model settings. Section 2.5 presents some simulation studies to illustrate the performance of our method. In Section 2.6, we apply the BSCC model to an ocean temperature and salinity data set. Section 2.7 concludes our method with some discussion. The proof of the main theoretical results, the detailed implementation and discussion of the RJ-MCMC algorithm, and additional simulation results are provided in the Appendix.

2.2 Related Work

A large body of model based spatial partition approaches have been proposed in various contexts. Methods such as Markov connected component fields (Gangnon and Clayton, 2000) and product partition models (Hegarty and Barry, 2008; Page and Quintana, 2016) take into account spatial information for clustering, but may not fully guarantee spatial contiguity or allow for arbitrary cluster shapes. Mixture models such as Dirichlet processes (e.g., Gelfand et al., 2005; Blei and Frazier, 2011; Zhang et al., 2014; Ma et al., 2020) are popular Bayesian nonparametric methods for clustering but tend to produce many small clusters. Space partitioning approaches, such as binary treed methods and Voronoi tessellations (Green and Sibson, 1978), have also been widely used in statistics to model responses locally in a region of the input space. Examples of binary treed methods include CART (Breiman et al., 1984; Chipman et al., 1998; Denison et al., 1998), BART (Chipman et al., 2010) and treed Gaussian processes (Gramacy and Lee, 2008; Konomi et al., 2014), where the input space is partitioned into non-overlapping regions by making binary splits recursively. On the other hand, Voronoi tessellation based models (e.g., Knorr-Held and Raßer, 2000; Denison and Holmes, 2001; Kim et al., 2005; Feng et al., 2016) define regions by a number of center locations such that points within a region are closer to its center than any other centers. However, both methods put considerable constraints on the shape of the regions. Voronoi tessellations imply a convexity assumption on the region shapes, and binary treed approaches only produce rectangle shaped regions. Spatial scan statistics (Kulldorff and Nagarwalla, 1995; Kulldorff, 1997) and their variants are also popular approaches to detect spatial clusters. Lin (2014) and Lin et al. (2016) consider Poisson regression models with spatially clustered intercepts using spatial scan statitstics. Lee et al. (2017) develop spatial cluster detection for regression coefficients using spatial scan statistics where the candidate clusters are often assumed to be circular windows.

Our method is motivated from Li and Sang (2019), who propose a fused lasso regularization and optimization method for spatially varying coefficient models, called the SCC, which uses a Euclidean distance based minimum spanning tree (MST) as the “spatial order" to encourage homogeneity between the regression coefficients at two adjacent locations. The method pursues a sparse solution on the difference between the two edge-connected coefficients, where the zero element indicates that two vertices belong to the same cluster, while the non-zero element corresponds to a cut set of edges which, if removed from the MST, will partition the vertices into a number of clusters. Nevertheless, the method does not produce uncertainty measures of parameter estimations. In addition, a fixed Euclidean MST is used as the spatial order for the regression coefficient of each covariate, which leads to over-clustering especially with small sample sizes as it only induces a restricted partition space to which the actual partition may not belong. In contrast, the Bayesian method developed in this work seeks to find the true spatial order by treating different spanning trees for each covariate as unknown parameters. We will show in Section 2.5 that this has a significant impact on the results, evidenced by the nearly 80% reduction in the mean square error of BSCC compared with that of SCC in simulation studies.

Most recently, Teixeira et al. (2015, 2019) also develop a Bayesian spatial partitioning model based on spanning trees for the clustering of spatial and spatial temporal responses, respectively. The idea is to construct a random partition model based on random spanning trees, where probabilistic prior models are assigned to the spanning trees and the edge removal probabilities. Their methods have shown a superior performance in terms of clustering accuracy for a number of spatial and spatial temporal clustering tasks, indicating a great potential of the random spanning tree methods. Following a similar spirit, the proposed model offers a new random spanning tree model which complements and differs from theirs in several main aspects. First, we extend beyond a single spanning tree partition model for spatial response data to a general hierarchical model setting for the multiple partitions of latent variables. Second, Teixeira et al. (2019) assume a uniform prior on the spanning tree space and an approximate sampler is used to sample a spanning tree in their MCMC algorithm. We overcome this issue by assigning uniform priors to edge weights in the original graph, which induces priors on the spanning tree space. An exact sampler based on this prior setting is proposed in this chapter. Third, they model the prior probability of a partition given a spanning tree by assigning a Beta-distributed prior on the edge inclusion probability without discussing the choice of its hyperparameters. We argue, from a theoretical point of view, that such choice needs careful considerations as it reflects penalty on the number of clusters and has profound effect on the asymptotic behavior of posterior distributions. In this work, we explicitly assign a penalized complexity prior on the number of partitions for which we prove the posterior consistency and design a tailored efficient RJ-MCMC algorithm. In addition, the posterior inference of their partitions relies on a pre-specified threshold of the edge inclusion probability, whereas our method allows us to directly obtain posterior samples of partitions. Finally, we derive a number of original non-asymptotic (e.g., Proposition 2) and asymptotic theories (e.g., Theorem 3), which provide a rigorous justification for the use of random spanning tree models.

2.3 Methodology

We begin with a varying coefficient regression model in the spatial context to illustrate our Bayesian partitioning method, and outline extensions to other hierarchical models with latent clustered variables in Section 2.4.

Let $[\{\mathbf{x}(\mathbf{s}_i), y(\mathbf{s}_i)\},\ i = 1, \ldots, n]$ be the spatial data observed at locations $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathcal{D} \subset \mathbb{R}^d$, where $\mathbf{x}(\mathbf{s}_i) = \{x_1(\mathbf{s}_i), \ldots, x_p(\mathbf{s}_i)\}^\mathsf{T} \in \mathbb{R}^p$ is a vector of covariates and $y(\mathbf{s}_i)$ is a scalar of response. We consider a model

$$y(\mathbf{s}_i) = \mathbf{x}^\mathsf{T}(\mathbf{s}_i)\boldsymbol{\beta}(\mathbf{s}_i) + \epsilon(\mathbf{s}_i), \quad \epsilon(\mathbf{s}_i) \overset{i.i.d.}{\sim} \mathrm{N}(0, \sigma^2), \tag{2.1}$$

where $\boldsymbol{\beta}(\mathbf{s}_i) = \{\beta_1(\mathbf{s}_i), \ldots, \beta_p(\mathbf{s}_i)\}^\mathsf{T}$ are unknown coefficients quantifying the relationships between the response and covariates, and $\epsilon(\mathbf{s}_i)$ are independently and identically distributed (i.i.d.) random noises. Clearly, this is a high-dimensional regression problem as there are n samples and np unknown regression coefficients. Assumptions need to be made on $\boldsymbol{\beta}(\mathbf{s}_i)$ to regularize this ill-posed problem. Previous spatial high-dimensional regression models often assume sparsity (Chu et al., 2011) or smoothness in $\boldsymbol{\beta}(\mathbf{s}_i)$ (Gelfand et al., 2003; Mu et al., 2018).

In this chapter, we are interested in detecting clustering patterns in $\boldsymbol{\beta}(\mathbf{s}_i)$. For each individual $\beta_m(\mathbf{s}_i)$ $(m = 1, \ldots, p)$, we assume there is a covariate-specific unknown disjoint partition such that $\beta_m(\mathbf{s}_i)$ is a spatially piecewise constant, i.e., $\beta_m(\mathbf{s}_i) = \beta_m(\mathbf{s}_j)$ if $\mathbf{s}_i$ and $\mathbf{s}_j$ are in the same cluster. Alternatively, one may assume there is a single common unknown partition for the whole vector $\boldsymbol{\beta}(\mathbf{s}_i)$, i.e., $\{\beta_1(\mathbf{s}_i), \ldots, \beta_p(\mathbf{s}_i)\}^\mathsf{T} = \{\beta_1(\mathbf{s}_j), \ldots, \beta_p(\mathbf{s}_j)\}^\mathsf{T}$ if $\mathbf{s}_i$ and $\mathbf{s}_j$ are in the same cluster. The advantage of the first assumption is that it allows us to make inference for the partition in each covariate. We adopt this assumption in this chapter since one may expect different cluster structures in coefficients for different covariates, but it is straightforward to extend our method to the second one.

In the Bayesian framework, we need to assign priors for the unknown partitions and to sample from the space of partitions for inference. In many spatial applications, as aforementioned, it is desired to consider partitions of locations with spatially contiguous clusters such that only adjacent locations are clustered together. When a complete order of regression coefficients is available, such as in time series problems (Kowal et al., 2019), we could obtain contiguous clusters easily by finding change points in the ordered coefficients. However, it is known that spatial data do not have a natural order. In this chapter, we propose to use spanning tree as the spatial order for cluster detection and by treating it as an unknown parameter, our method can adaptively learn the spanning tree order and detect changes in the tree-ordered coefficients.

Below, we give formal definitions for contiguous partitions and clusters, and construct a spanning tree model for such partitions.

2.3.1 A Prior Model for Contiguous Partitions

Consider an undirected graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$, where $\mathcal{V}_0 = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ is the vertex set and the edge set $\mathcal{E}_0$ is a subset of $\{(\mathbf{s}_i, \mathbf{s}_j) : \mathbf{s}_i, \mathbf{s}_j \in \mathcal{V}_0, \mathbf{s}_i \neq \mathbf{s}_j\}$. Note that in $\mathcal{E}_0$, $(\mathbf{s}_i, \mathbf{s}_j)$ is an unordered pair. Given a spatial data set, we can construct an undirected graph $\mathcal{G}_0$ to represent the relationship of spatial adjacency or neighborhood. For regularly spaced data, a lattice graph is a common choice. For irregularly spaced data, one straightforward way for construction is to connect a vertex with all its neighbors within a certain radius. Another approach is the Delaunay triangulation (Lee and Schachter, 1980), which constructs triangles

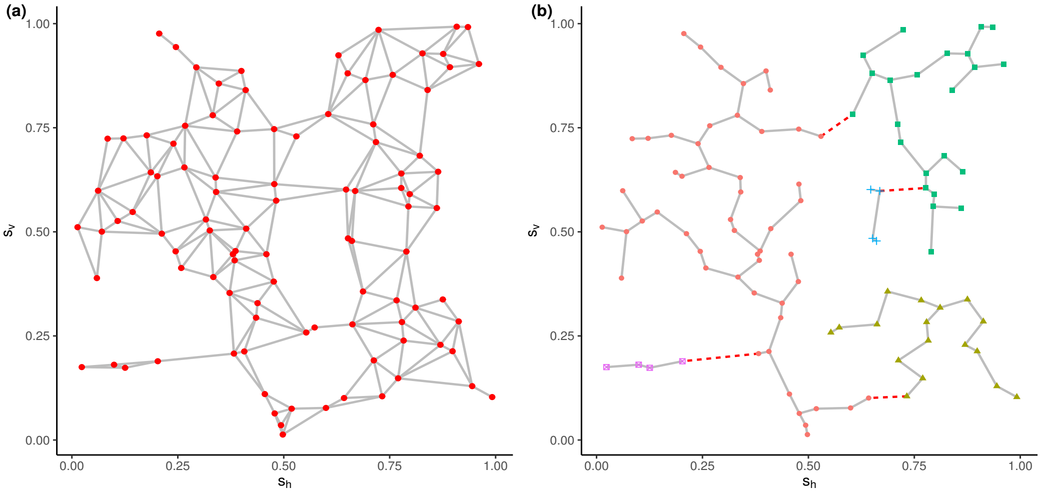

Figure 2.1: (a) A graph constructed by the Delaunay triangulation, with edges longer than 0.2 removed. (b) An example of a partition with 5 clusters induced by removing the set of red dashed edges from a spanning tree of the graph in (a). Different clusters are marked by different colors.

with a vertex set $\mathcal{V}_0$ such that no vertex is inside the circumcircle of any triangle. In practice, edges longer than a certain threshold are removed to ensure spatial proximity of neighboring vertices. Figure 2.1(a) demonstrates an example of the Delaunay triangulation. We will show in Section 2.3.3 that spatial graphs constructed by these two approaches achieve nice theoretical properties.

In graph theory, a sequence of edges $\{(\mathbf{s}_{i_0}, \mathbf{s}_{i_1}), \ldots, (\mathbf{s}_{i_{t-1}}, \mathbf{s}_{i_t})\} \subseteq \mathcal{E}_0$ is called a path of length t between $\mathbf{s}_{i_0}$ and $\mathbf{s}_{i_t}$ if all $\mathbf{s}_{i_j}$'s are distinct. It is called a cycle if $\mathbf{s}_{i_0} = \mathbf{s}_{i_t}$ and all other vertices are distinct. A graph $\mathcal{G}_0$ is said to be connected if for any two vertices there exists a path between them. In this chapter we assume $\mathcal{G}_0$ is always connected. A subgraph $(\mathcal{V}, \mathcal{E}), \mathcal{V} \subseteq \mathcal{V}_0, \mathcal{E} \subseteq \mathcal{E}_0$ is called a connected component of $\mathcal{G}_0$ if it is connected and there is no path between any vertex in $\mathcal{V}$ and any vertex in $\mathcal{V}_0 \setminus \mathcal{V} := \{\mathbf{s} \in \mathcal{V}_0 : \mathbf{s} \notin \mathcal{V}\}$, the difference between sets $\mathcal{V}_0$ and $\mathcal{V}$. Now one can define spatially contiguous partitions and clusters formally based on the notion of connected components (Teixeira et al., 2019).

Definition 2.1. Given an undirected graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$, a subset $\mathcal{C} \subseteq \mathcal{V}_0$ is a spatially contiguous cluster if there exists a connected subgraph $(\mathcal{C}, \mathcal{E}_\mathcal{C}), \mathcal{E}_\mathcal{C} \subseteq \mathcal{E}_0$. A spatially contiguous partition of $\mathcal{V}_0$ is a collection of disjoint spatially contiguous clusters $\pi = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ such that $\cup_{j=1}^{k} \mathcal{C}_j = \mathcal{V}_0$.

For conciseness, henceforth, we refer to spatially contiguous partitions and clusters as partitions and clusters, respectively. Our goal is to develop a partition model for a given spatial graph. However, it is a long-standing challenging task since the number of all possible partitions grows rapidly as the number of locations. Following the similar ideas as in Teixeira et al. (2015, 2019) and Li and Sang (2019), we consider a much more compact representation of spatially contiguous partitions based on spanning trees.

A spanning tree of a graph $\mathcal{G}_0$ is defined as a subgraph $\mathcal{T} = (\mathcal{V}_0, \mathcal{E}_\mathcal{T}), \mathcal{E}_\mathcal{T} \subseteq \mathcal{E}_0$ that connects all vertices without any cycle. Therefore, a spanning tree has $|\mathcal{V}_0|$ vertices and $|\mathcal{V}_0| - 1$ edges, where $|\mathcal{V}_0|$ denotes the cardinality of set $\mathcal{V}_0$. By definition, there can be multiple spanning trees for a given graph. Suppose that weights w_e are assigned to each edge $e \in \mathcal{E}_0$, and then an MST is a spanning tree $(\mathcal{V}_0, \mathcal{E}_\mathcal{T}), \mathcal{E}_\mathcal{T} \subseteq \mathcal{E}_0$ that has the minimal sum of weight $\sum_{e \in \mathcal{E}_\mathcal{T}} w_e$.

A partition with $k+1$ clusters can also be defined by a spanning tree and a subset of edges $\mathcal{E}_k \subseteq \mathcal{E}_\mathcal{T}$ of cardinality k. Specifically, as shown in Figure 2.1(b), if a set of k edges is removed from a spanning tree $\mathcal{T}$, we create a subgraph of $\mathcal{T}$ that has $k+1$ connected components, and the vertex set of each component forms a cluster. Throughout this chapter, we say a partition is *induced* by a spanning tree $\mathcal{T}$ if the partition can be obtained by removing a subset of edges from $\mathcal{E}_\mathcal{T}$.

Below, we show the sample space of partitions induced from random spanning trees accommodates all possible contiguous partitions.

Proposition 2.2. *Let $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ be a connected graph and $\pi = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ be an arbitrary spatially contiguous partition of $\mathcal{V}_0$. There exists at least one spanning tree $\mathcal{T} = (\mathcal{V}_0, \mathcal{E}_\mathcal{T}), \mathcal{E}_\mathcal{T} \subseteq \mathcal{E}_0$ and a subset $\mathcal{E}_{k-1} \subseteq \mathcal{E}$ of cardinality $k-1$ that induce π.*

Proposition 2.2 implies that we can represent any partition by a spanning tree and a subset of its edge set. It is notable that there is no assumption on the shape and size of each cluster in the partition. The detailed proof of Proposition 2.2 is provided in Appendix A.1.1.

The above discussion suggests that the prior model specification for partitions boils down to assigning prior models for spanning trees and the removed edge set given a spanning tree. Conditional on a spanning tree $\mathcal{T}$ and the number of clusters k, we can impose a prior on the space of partitions induced by the spanning tree, or equivalently, on the selection of $(k-1)$-sized subsets of $\mathcal{E}_{\mathcal{T}}$. Then we can assign a prior on the space of all possible spanning trees and a prior on the number of clusters.

Formally, let $\mathcal{T}^{(m)}$ be a spanning tree of $\mathcal{G}_0$ that can induce $\pi^{(m)}$, the partition associated with the mth covariate. Conditional on $\mathcal{T}^{(m)}$ and k_m, we assume independent uniform priors on all possible $\pi^{(m)}$'s with k_m clusters that are induced by $\mathcal{T}^{(m)}$ (also see Teixeira et al. 2015, 2019 for an alternative prior model on partitions):

$$p\left\{\pi^{(m)} \mid k_m, \mathcal{T}^{(m)}\right\} \propto \mathbf{1}\{\pi^{(m)} \text{ is induced by } \mathcal{T}^{(m)} \text{ and has } k_m \text{ clusters}\}, \tag{2.2}$$

independently for $m = 1, \dots, p$, where $\mathbf{1}(\cdot)$ is an indicator function. From the perspective of variable selection, our prior is equivalent to assigning equal probability to all possible selections of $k_m - 1$ edges from the edge set of size $n-1$.

To specify the prior on $\mathcal{T}^{(m)}$, we let $\mathbf{w}^{(m)} = \{w_{ij}^{(m)}\}_{(\mathbf{s}_i,\mathbf{s}_j)\in\mathcal{E}_0}$ be a vector of edge weights associated with the mth covariate, where $w_{ij}^{(m)}$ is the weight for edge $(\mathbf{s}_i, \mathbf{s}_j)$. We assign independent and identical Unif $(0,1)$ prior on $w_{ij}^{(m)}$ and let $\mathcal{T}^{(m)}$ be the MST given $\mathbf{w}^{(m)}$, i.e.,

$$\mathcal{T}^{(m)} = \mathrm{MST}\{\mathbf{w}^{(m)}\}, \quad w_{ij}^{(m)} \overset{i.i.d.}{\sim} \text{Unif}\,(0,1), \tag{2.3}$$

where MST($\mathbf{w}$) means an MST of the graph $\mathcal{G}_0$ based on edge weights $\mathbf{w}$ given by Prim's algorithm. Recall that an MST is a spanning tree that has minimal sum of edge weights and it is determined by the edge weights of the original graph. Also note that for any given spanning tree of the original graph, there exists a set of edge weights such that the resulting

MST produces that spanning tree. Therefore, the prior on edge weights induces a prior model on the resulting spanning tree. Note, however, that our induced prior on the space of spanning trees is not uniform, in contrast to the prior in Teixeira et al. (2015, 2019), who use an approximate sampler to update spanning trees. Our prior setting leads to an *exact* update of $\mathcal{T}^{(m)}$ in our RJ-MCMC algorithm (see Section 2.3.4 for details).

Finally, we assign the following prior to the number of clusters for each coefficient, following the setup of Knorr-Held and Raßer (2000) and Feng et al. (2016):

$$\mathbb{P}(k_m = k) \propto (1-c)^k, \quad \text{for } k = 1, \ldots, n, \ 0 \le c < 1 \tag{2.4}$$

independently for all m. This prior is a geometric distribution truncated to the support $\{1, \ldots, n\}$ with prior mean $\mathbb{E}(k_m) = 1/c - n(1-c)^n/\{1-(1-c)^n\}$ when $0 < c < 1$; when $c = 0$ the prior becomes a truncated discrete uniform distribution with prior mean $\mathbb{E}(k_m) = (1+n)/2$. It is noted that this prior has a geometrically decaying probability with hyperparameter c controlling the decaying rate, and hence serves as a prior to penalize the model with a large number of clusters. If c is closer to 1 we have a stronger penalization for the large number of clusters. The choice of c plays a crucial role in high-dimensional settings. We will show in Section 2.3.3 that a theoretically viable choice is to let $-\log(1-c)$ grow at the same rate as $\log(|\mathcal{V}_0|)$. It is possible to assign a prior on k_m conditional on $\mathcal{T}^{(m)}$; however, when there is no *a priori* information about the true partitions and the spanning trees that induce them, we assume that the priors for k_m are independent of $\mathcal{T}^{(m)}$.

2.3.2 Bayesian Hierarchical Spatially Clustered Coefficient Models

Let $\pi^{(m)} = \{\mathcal{C}_1^{(m)}, \ldots, \mathcal{C}_{k_m}^{(m)}\}$ $(m = 1, \ldots, p)$ be the spatial partition of the regression coefficient associated with the mth covariate, $\boldsymbol{\beta}^{(m)} = \{\beta_1^{(m)}, \ldots, \beta_{k_m}^{(m)}\}^\intercal$ be the vector of all different values of the mth coefficient, where $\beta_j^{(m)}$ is the coefficient value associated with cluster $\mathcal{C}_j^{(m)}$. With a slight abuse of notation, we denote $\mathbf{s}_i \in \mathcal{C}_{j_1}^{(1)} \cap \cdots \cap \mathcal{C}_{j_p}^{(p)}$ for some $j_1, \ldots, j_p$, if the regression coefficient at $\mathbf{s}_i$ for the mth covariate belongs to $\mathcal{C}_{j_m}^{(m)}$. Choosing conjugate priors for other model parameters, our hierarchical model can be written as

$$y(\mathbf{s}_i) \mid \{\boldsymbol{\beta}^{(m)}\}_{m=1}^{p}, \sigma^2, \lambda, \left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p} \overset{ind.}{\sim} \mathrm{N}\left\{\sum_{m=1}^{p} \beta_{j_m}^{(m)} x_m(\mathbf{s}_i),\ \sigma^2\right\}, \tag{2.5a}$$

$$\boldsymbol{\beta}^{(m)} \mid \sigma^2, \lambda, \pi^{(m)}, k_m \overset{ind.}{\sim} \mathrm{N}_{k_m}\left(\mathbf{0}, \lambda^{-1}\sigma^2\boldsymbol{\Sigma}_m\right), \tag{2.5b}$$

$$\left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p} \sim \prod_{m=1}^{p} p\left\{\pi^{(m)} \mid k_m, \mathbf{w}^{(m)}\right\} p(k_m) p\{\mathbf{w}^{(m)}\}, \tag{2.5c}$$

$$\sigma^2 \sim \mathrm{IG}(a_0/2, b_0/2), \tag{2.5d}$$

$$\lambda \sim \mathrm{Gamma}(c_0/2, d_0/2), \tag{2.5e}$$

where N_{k_m} represents the k_m-dimensional multivariate normal distribution, $\boldsymbol{\Sigma}_m$ is a $k_m \times k_m$ covariance matrix, $\mathrm{IG}(a, b)$ is the inverse-Gamma distribution, $\mathrm{Gamma}(a, b)$ is the Gamma distribution in shape-rate parameterization, and a_0, b_0, c_0, d_0 are hyperparameters. The notation "*ind.*" means that we assume (2.5a) holds independently for all $i = 1, \ldots, n$ and place independent prior (2.5b) on $\boldsymbol{\beta}^{(m)}$ for all $m = 1, \ldots, p$. The priors in (2.5c), (2.5d), and (2.5e) are also assumed to be mutually independent. We allow the prior of $\boldsymbol{\beta}^{(m)}$ to accommodate spatial dependence among clusters if one assumes spatial structures in $\boldsymbol{\Sigma}_m$. In the case where there is no prior information on the spatial dependence structure of $\boldsymbol{\beta}^{(m)}$, one can set $\boldsymbol{\Sigma}_m = \mathbf{I}_{k_m}$, the $k_m \times k_m$ identity matrix. We only consider this independent case in this chapter for simplicity. Note that it is also possible to choose other priors for $\boldsymbol{\beta}^{(m)}$, σ^2, and λ. For example, one can place non-informative priors on σ^2 and λ. And we specify independent and identical priors for the partitions of each regression coefficient, $\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\}$, following the method described in Section 2.3.1.

2.3.3 Theoretical Properties

To ease notations, we present our theoretical results for $p = 1$ case,

$$y(\mathbf{s}_i) = x(\mathbf{s}_i)\beta(\mathbf{s}_i) + \epsilon(\mathbf{s}_i),$$

where $x(\mathbf{s}_i), \beta(\mathbf{s}_i) \in \mathbb{R}$, though the result can be extended to a more general case. In this subsection, we let x_i and β_i denote $x(\mathbf{s}_i)$ and $\beta(\mathbf{s}_i)$, respectively. Let $\boldsymbol{\beta} = (\beta_1, \dots, \beta_n)^\mathsf{T}$. Given a spanning tree $\mathcal{T} = (\mathcal{V}_0, \mathcal{E}_\mathcal{T})$, we define $G^*_\mathcal{T} = \{(\mathbf{s}_i, \mathbf{s}_j) \in \mathcal{E}_\mathcal{T} : \beta^*_i - \beta^*_j \neq 0\}$, where β^*_i is the true value of β_i with the corresponding true partition denoted as π^*. We assume that the number of clusters in π^*, denoted by k^*, is *fixed*. $G^*_\mathcal{T}$ represents the edges of $\mathcal{T}$ that have nonzero jumps in $\boldsymbol{\beta}^*$, the true value of $\boldsymbol{\beta}$. When π^* is induced by $\mathcal{T}$ so that there is exactly one jump in $\mathcal{E}_\mathcal{T}$ that crosses two distinct clusters, $|G^*_\mathcal{T}| + 1$ equals k^*. Otherwise, $|G^*_\mathcal{T}|$ will be larger than $k^* - 1$. Indeed, in this case, we get a nested partition of the true π^* when $G^*_\mathcal{T}$ is removed from $\mathcal{E}_\mathcal{T}$. We let $\mathbb{T}_n$ be the set of all spanning trees of the graph $\mathcal{G}_0$ with n vertices, and define $g^*_n = \max_{\mathcal{T} \in \mathbb{T}_n} |G^*_\mathcal{T}| + 1$ such that $g^*_n - 1$ is the maximum number of edges that have nonzero jumps in $\boldsymbol{\beta}^*$ among all possible spanning trees.

We adopt the following asymptotic notations. Given two positive sequences $\{a_n\}$ and $\{b_n\}$, $a_n \succ b_n$ means $\lim_{n\to\infty} (a_n/b_n) = \infty$ and $a_n \asymp b_n$ means $0 < \liminf_{n\to\infty} (a_n/b_n) \leq \limsup_{n\to\infty} (a_n/b_n) < \infty$. We also denote the L_2 norm by $\|\cdot\|$.

Our results on posterior consistency rely on the following assumptions as $n \to \infty$:

(C1) x_i is non-random, and $|x_i| \leq M_0$ for some $M_0 > 0$ and any i.

(C2) $\log\left(\max_{1\leq i\leq n} |\beta^*_i|/\sigma^*\right) = O(\log n)$, where σ^* is the fixed true value of σ as n grows.

(C3) The graph satisfies $g^*_n \prec n/\log n$. Let P_n be the number of all unique partitions nested in π^* that have at most $g^*_n q_n$ clusters for a given sequence $q_n \to \infty$. We assume that $\log P_n = O(g^*_n \log n)$.

(C4) $1 - c \asymp n^{-\alpha}$ for some constant $\alpha > 0$.

Assumption (C1) is a commonly adopted assumption which states that the covariate space is bounded. Assumption (C2) constrains the asymptotic growth rate of the magnitude of the true coefficients (see, e.g., Song and Cheng, 2020). Assumption (C3) restricts the number of edges that have nonzero jumps in $\boldsymbol{\beta}^*$ for any possible spanning tree, and essentially excludes graphs that are too dense. We will show that $g_n^* \prec n/\log n$ is satisfied by commonly used spatial designs and graphs with probability tending to 1 in Proposition 2.5. The second part of Assumption (C3) constrains the complexity of the space of partitions to ensure the existence of test functions in our proof. Assumption (C4) imposes restriction on the tail behavior of our penalized complexity prior such that it provides enough probability mass around the true model. Similar conditions on prior hyperparameters are common in Bayesian high-dimensional regression literature (see, e.g., Armagan et al., 2013; Yang et al., 2016).

The following theorem states that if Assumptions (C1)-(C4) hold, the posterior distribution of the predicted responses from BSCC model concentrates around the true means asymptotically.

Theorem 2.3. *(Posterior consistency for fixed spatial graph designs) Let $\boldsymbol{\mu}$ and $\boldsymbol{\mu}^*$ be n-dimensional vectors such that $\mu_i = x_i\beta_i$ and $\mu_i^* = x_i\beta_i^*$. Under Assumptions (C1)-(C4), there exists a constant $M_1 > 0$ and $\varepsilon_n \asymp \sqrt{g_n^* \log n/n}$ such that the posterior distribution satisfies*

$$\Pi_n\left(\frac{1}{\sqrt{n}}\|\boldsymbol{\mu}-\boldsymbol{\mu}^*\| \geq M_1\sigma^*\varepsilon_n \mid \mathbf{y}\right) \longrightarrow 0$$

with probability one.

The detailed proof is provided in Appendix A.1.2.

We verify that the first part of Assumption (C3) holds with probability tending to 1 for some common choices of spatial designs and spatial graphs. In the spatial context, we consider an asymptotic framework for piecewise constant functions that are defined on spatial random graphs with a diverging number of vertices in $\mathbb{R}^2$. Before giving the proposition, we will first describe a formulation for the sampling region and a nonuniform random spatial design for irregularly spaced data, and then a technical definition of piecewise constant functions will be introduced.

Below, we state assumptions on the sampling region $\mathcal{D}$ and the sampling design of n points $\mathbf{s}_1^D, \cdots, \mathbf{s}_n^D$ in $\mathcal{D}$.

(C5) *Spatial sampling region.* Assume $\mathcal{D}$ is homeomorphic to the unit square with the Euclidean metric and a bi-Lipschitz homeomorphism $\mathcal{F}_D : \mathcal{D} \to [0,1]^2$. Under this assumption, $\mathcal{S}_n = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n)$, where $\mathbf{s}_i = \mathcal{F}_D(\mathbf{s}_n^D)$ for $i = 1, \cdots, n$ is the mapping of the original sampling point to $[0,1]^2$. This condition allows us to consider a study region with a variety of shapes as long as it is topologically equivalent to a unit square.

(C6) *Spatial design and spatial graph.* Given $n \in \mathbb{N}$, we assume $\mathcal{S}_n$ is a sequence of n independent points where each point is distributed on $[0,1]^2$ with a probability density function p_s such that $0 < p_s^{\min} \leq p_s(\mathbf{s}) \leq p_s^{\max} < \infty$. We assume the spatial graph on $\mathcal{S}_n$ is constructed by (i) the radius-based nearest neighbor (R-NN) graph with a radius $\gamma_1 \asymp \sqrt{\log n/n}$ and $\gamma_1 > \gamma_0$, where γ_0 is the maximum edge length of the MST on $\mathcal{S}_n$; or (ii) the Delaunay triangulation graph where the edges are removed if they are longer than γ_2, where $\gamma_2 \asymp \sqrt{\log n/n}$ and $\gamma_2 > \gamma_0$. We will refer to it as the *restricted Delaunay triangulation* in the proof.

Notice that $|G_{\mathcal{T}}^*|$ is essentially the number of edges across the cluster boundaries of the true coefficient, which is viewed as a piecewise constant function defined on the spatial domain $\mathcal{D}$. To bound $\max_{\mathcal{T} \in \mathbb{T}_n} |G_{\mathcal{T}}^*|$, we work with the following definition of piecewise constant functions, in which the cluster boundary set is introduced.

Definition 2.4. (see, e.g., Willett et al. 2006) We say that a function $g : [0,1]^2 \to \mathbb{R}$ is *piecewise constant* if there exists a cluster boundary set $\mathcal{B}_g$ such that:

1. The cluster boundary set $\mathcal{B}_g$ has a υ_n-covering number $N(\mathcal{B}_g, \upsilon_n, \|\cdot\|) \leq M_2 \upsilon_n^{-1}$, for some constant $M_2 > 0$.

2. The function g is locally constant on $[0,1]^2 \backslash \mathcal{B}_g$, i.e., $g(\mathbf{s}) = g(\mathbf{s}')$ if $\mathbf{s}$ and $\mathbf{s}'$ belong to the same connected component of $[0,1]^2 \backslash \mathcal{B}_g$.

The next proposition states that the condition $g_n^* \prec n/\log n$ is met under Assumption (C5) and (C6) with high probability. The proof is delayed to Appendix A.1.3.

Proposition 2.5. *Assume further the true regression coefficient* $\beta^{*,D}$ *is a function* $\beta^{*,D}(\mathbf{s}^D) : \mathcal{D} \to \mathbb{R}$ *such that* $\beta^*(\mathbf{s}) : [0,1]^2 \to \mathbb{R}$ *is piecewise-constant on* $[0,1]^2$ *with the boundary set* $\mathcal{B}_{\beta^*}$*. Under Assumptions (C5) and (C6), there exist positive constants* $M_3, M_4 > 0$*, such that* $g_n^* \leq M_3\sqrt{n \log n}$ *holds with probability at least* $1 - \exp\left(-M_4\sqrt{n \log n}\right)$*.*

Combining Theorem 2.3 and Proposition 2.5 gives the following posterior concentration result under the random spatial graph in Assumption (C6). The proof is given in Appendix A.1.4.

Corollary 2.6. *(Posterior consistency for random spatial graph designs) Let* $\tilde{P}_n$ *be the number of all unique partitions nested in* π^* *that have at most* $M_3 q_n \sqrt{n \log n}$ *clusters, where* π^* *is the true partition corresponding to* $\beta^*(\mathbf{s})$ *in Proposition 2.5 given* $\mathcal{S}_n$*. Assume that* $\log \tilde{P}_n \leq M_5 n^{1/2} \log^{3/2} n$ *with probability tending to one for some constant* $M_5 > 0$ *not depending on* $\mathcal{S}_n$*. Under Assumptions (C1), (C2) and (C4)-(C6), there exists a constant* $M_6 > 0$ *and* $\tilde{\varepsilon}_n \asymp n^{-1/4} \log^{3/4} n$ *such that the posterior distribution satisfies*

$$\Pi_n\left(\frac{1}{\sqrt{n}}\|\boldsymbol{\mu} - \boldsymbol{\mu}^*\| \geq M_6 \sigma^* \tilde{\varepsilon}_n \mid \mathbf{y}, \mathcal{S}_n\right) \longrightarrow 0$$

in probability.

2.3.4 Computational Strategies

We extend conventional RJ-MCMC algorithm to sample the partitions, the values of coefficients, and other parameters simultaneously. Standard RJ-MCMC algorithm may suffer from poor mixing and slow convergence, because of the potentially multimodal posterior (which is common in many partition models such as Chipman et al. 1998) and the large space of spanning trees. We propose several strategies to address computational issues.

Let $\mathbf{y} = \{y(\mathbf{s}_1), \ldots, y(\mathbf{s}_n)\}^\mathsf{T}$ be the vector of responses, $\tilde{\boldsymbol{\beta}} = [\{\boldsymbol{\beta}^{(1)}\}^\mathsf{T}, \ldots, \{\boldsymbol{\beta}^{(p)}\}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^K$ be the stacked vector of coefficients, where $K = \sum_{m=1}^p k_m$, and $\widetilde{\mathbf{X}} = [\widetilde{\mathbf{X}}_1 \cdots \widetilde{\mathbf{X}}_p] \in \mathbb{R}^{n \times K}$ be the design matrix associated with $\tilde{\boldsymbol{\beta}}$, where each sub-matrix $\widetilde{\mathbf{X}}_m \in \mathbb{R}^{n \times k_m}$ is constructed in the following way. The (i,j)th element of $\widetilde{\mathbf{X}}_m$ is set to be $x_m(\mathbf{s}_i)$ if the ith location belongs to cluster $\mathcal{C}_j^{(m)}$ for some $j \in \{1,\dots,k_m\}$; otherwise, it is set to be zero.

We first rewrite the data model and the prior model for $\tilde{\boldsymbol{\beta}}$ in matrix forms as

$$\begin{aligned} \mathbf{y} \mid \tilde{\boldsymbol{\beta}}, \sigma^2, \lambda, \left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p} &\sim \mathrm{N}_n\left(\widetilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}, \sigma^2\mathbf{I}_n\right) \\ \tilde{\boldsymbol{\beta}} \mid \sigma^2, \lambda, \left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p} &\sim \mathrm{N}_K\left(\mathbf{0}, \lambda^{-1}\sigma^2\mathbf{I}_K\right) \end{aligned}$$

Integrating out $\tilde{\boldsymbol{\beta}}$, the marginal distribution of $\mathbf{y}$ becomes

$$\mathbf{y} \mid \sigma^2, \lambda, \left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p} \sim \mathrm{N}_n\left(\mathbf{0}, \sigma^2\mathbf{P}_\lambda\right), \tag{2.6}$$

where $\mathbf{P}_\lambda = \mathbf{I}_n + \lambda^{-1}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\intercal$. It allows us to sample from the collapsed posterior distribution of $\left[\left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p}, \sigma^2, \lambda\right]$ as follows

$$\begin{aligned} & p\left[\left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p}, \sigma^2, \lambda \;\middle|\; \mathbf{y}\right] \propto \\ & \qquad (\sigma^2)^{-n/2}|\mathbf{P}_\lambda|^{-1/2}\exp\left(-\frac{1}{2\sigma^2}\mathbf{y}^\intercal\mathbf{P}_\lambda^{-1}\mathbf{y}\right)\cdot(\sigma^2)^{-a_0/2-1}\exp\left(-\frac{b_0}{2\sigma_y^2}\right)\times \\ & \qquad\qquad \lambda^{c_0/2-1}\exp\left(-\frac{d_0}{2}\lambda\right)\cdot\prod_{m=1}^{p}\left\{\binom{n-1}{k_m-1}^{-1}\cdot(1-c)^{k_m}\right\}. \end{aligned} \tag{2.7}$$

Standard uncollapsed MCMC can lead to poor mixing due to the strong dependence of $\tilde{\boldsymbol{\beta}}$. This collapsed posterior greatly improves the efficiency and mixing in searching the posterior of partitions.

Since the number of clusters in each partition is unknown, we employ the reversible jump MCMC (Green, 1995) to sample from the posterior in (2.7). Within each iteration of RJ-MCMC, we further iterate through each covariate from $m = 1$ to p. In each inner iteration, one of the following four possible moves is performed.

(a) *Birth*: Fixing the spanning tree $\mathcal{T}^{(m)}$, add a new cluster to $\pi^{(m)}$ by splitting an existing cluster.

(b) *Death*: Fixing $\mathcal{T}^{(m)}$, randomly remove an existing cluster by merging it into an adjacent cluster.

(c) *Change*: Fixing $\mathcal{T}^{(m)}$, randomly remove an existing cluster by merging it into an adjacent cluster, and then add a new cluster by splitting an existing cluster, so that the number of clusters remains unchanged.

(d) *Hyper*: Update parameters σ^2, λ, and $\mathbf{w}^{(m)}$ (and hence $\mathcal{T}^{(m)}$). Specifically, σ^2 is updated by a Gibbs step, $\mathbf{w}^{(m)}$ is updated by sampling a set of edge weights such that the resulting MST can induce the current sample of $\pi^{(m)}$ using an *exact* algorithm derived below, and λ is updated by a Metropolis-Hastings procedure with a symmetric random walk proposal.

The exact update of $\mathcal{T}^{(m)}$ is done by a Metropolis-Hastings algorithm to sample edge weights followed by Prim's algorithm. From (2.7) we have the full conditional of $\mathbf{w}^{(m)}$ proportional to

$$\mathbf{1}\left[\pi^{(m)} \text{ is induced by } \mathrm{MST}\{\mathbf{w}^{(m)}\} \text{ and } 0 < w_{ij}^{(m)} < 1 \text{ for all } (\mathbf{s}_i, \mathbf{s}_j) \in \mathcal{E}_0\right]. \tag{2.8}$$

We propose a new $\mathbf{w}^{(m)}$ by sampling $w_{ij}^{(m)}$ from i.i.d. Unif $(1/2, 1)$ if $\mathbf{s}_i$ and $\mathbf{s}_j$ are in different clusters and sampling $w_{ij}^{(m)}$ from i.i.d. Unif $(0, 1/2)$ if $\mathbf{s}_i$ and $\mathbf{s}_j$ are in a same cluster. The resulting spanning tree from Prim's algorithm based on the proposed edge weights is guaranteed to induce the current partition $\pi^{(m)}$ (Teixeira et al., 2015). The acceptance probability for $\mathbf{w}^{(m)}$ is always 1. To see this, first notice that (2.8) remains the same for the proposed weights, and thus the likelihood ratio is 1. The prior ratio is also 1 since we assume a uniform prior on $\mathbf{w}^{(m)}$. Due to the design of proposal distribution, the proposal ratio is again 1 as the sets of cross-cluster edges and within-cluster edges are preserved. The sample of $\mathcal{T}^{(m)}$ is the MST generated by Prim's algorithm. Note that this sampler is exact in the sense that there is no approximation in this sampling scheme. The induced chain of spanning trees is irreducible, as suggested by the following proposition.

Algorithm 1: RJ-MCMC algorithm

```
Initialize partitions and edge weights $\left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p}$;
for $t \leftarrow 1$ to $T$ do
    for $m \leftarrow 1$ to $p$ do
        Propose a birth, death, change, or hyper step with certain probabilities ;
        if birth step then
            Propose a new cluster by splitting an existing cluster in $\pi^{(m)}$ ;
        else if death step then
            Randomly remove an existing cluster by merging it to a neighboring
              cluster in $\pi^{(m)}$ ;
        else if change step then
            Randomly remove an existing cluster by merging it to a neighboring
              cluster, then propose a new cluster by splitting an existing cluster ;
        else if hyper step then
            Update $\sigma^2$ using Gibbs sampling ;
            Update $\mathbf{w}^{(m)}$ (and hence $\mathcal{T}^{(m)}$) by a Metropolis-Hastings step ;
            Update $\lambda$ by a Metropolis-Hastings step ;
        Accept proposed change with probability $\alpha_1$;
Discard samples from burn-in period;
```

Proposition 2.7. *For any spanning tree $\mathcal{T}$ of $\mathcal{G}_0$ that induces a partition π, the spanning tree sampling algorithm described above generates $\mathcal{T}$ with strictly positive probability.*

The proof of Proposition 2.7 is postponed to Appendix A.1.5.

We set the probability for each move to be $r_B(k) = 0.425, r_D(k) = 0.425, r_C(k) = 0.1$, and $r_H(k) = 0.05$, respectively. Adjustments are made for boundary cases when $k_m = 1$ or n. The choice of these probabilities works well empirically in our studies. But we remark that these probabilities can be modified if desired. For the first three moves, a new partition is accepted with probability $\alpha_1 = \min(1,\ \mathcal{A} \cdot \mathcal{P} \cdot \mathcal{L})$, where $\mathcal{A}, \mathcal{P}, \mathcal{L}$ are the prior ratio, proposal ratio, and likelihood ratio, respectively. For the fourth move, hyper, the spanning tree is updated adaptively to the current estimate of the partition, thus allowing for the search of spanning trees that can induce the true partitions. The RJ-MCMC algorithm is summarized in Algorithm 1 and detailed in Appendix A.2.

After obtaining samples of $\boldsymbol{\theta} = \left[\left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p}, \sigma^2, \lambda\right]$, it is straightforward to obtain a sample of $\tilde{\boldsymbol{\beta}}$ by sampling from $p(\tilde{\boldsymbol{\beta}} \mid \boldsymbol{\theta}, \mathbf{y})$, which takes the following closed form

$$\tilde{\boldsymbol{\beta}} \mid \boldsymbol{\theta}, \mathbf{y} \sim \mathrm{N}_K \left\{ (\widetilde{\mathbf{X}}^\mathsf{T}\widetilde{\mathbf{X}} + \lambda \mathbf{I}_K)^{-1}\widetilde{\mathbf{X}}^\mathsf{T}\mathbf{y},\ \sigma^2(\widetilde{\mathbf{X}}^\mathsf{T}\widetilde{\mathbf{X}} + \lambda \mathbf{I}_K)^{-1} \right\}.$$

One computation bottleneck is the evaluation of the likelihood function in (2.6), which involves the inversion of the $n \times n$ matrix $\mathbf{I}_n + \lambda^{-1}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\mathsf{T}$. Recall that the dimension of $\widetilde{\mathbf{X}}$ is $n \times K$, where K is the summed number of clusters over all covariates. As K is typically much smaller than n, we take advantage of the low-rank structure and apply the Sherman-Woodbury-Morrison formula to reduce the problem to computing $\mathbf{y}^\mathsf{T}\widetilde{\mathbf{X}}(\lambda \mathbf{I}_K + \widetilde{\mathbf{X}}^\mathsf{T}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^\mathsf{T}\mathbf{y}$.

The update of the above quadratic form in each MCMC iteration can be further simplified by the fact that most columns of $\widetilde{\mathbf{X}}$ are unchanged in a birth, death, or change step. For instance, in a birth step, $\widetilde{\mathbf{X}}$ is changed by adding one column and modifying another, which can be done by removing one column and adding two. The Cholesky decomposition of $\lambda \mathbf{I}_K + \widetilde{\mathbf{X}}^\mathsf{T}\widetilde{\mathbf{X}}$ can therefore be updated efficiently from the Cholesky factor at the previous step following the supernodal sparse Cholesky update/downdate algorithms (Chen et al., 2008; Osborne, 2010). $\widetilde{\mathbf{X}}^\mathsf{T}\mathbf{y}$ can also be updated by changing one element and adding/removing another. The overall time complexity to update the quadratic term is $O(nK)$, whereas directly evaluating it requires $O(nK^2)$ operations.

Finally, it is common to have multimodal posterior distributions for some parameters near cluster boundaries. We employ parallel tempering (Geyer, 1991) to better explore the posterior and improve mixing. Specifically, we run d chains in parallel with the likelihood function tempered by different "temperatures". The target distribution of the jth chain is

$$p_j(\boldsymbol{\theta} \mid \mathbf{y}) \propto \{\ell(\boldsymbol{\theta} \mid \mathbf{y})\}^{\nu_j}\, p(\boldsymbol{\theta}),$$

where $\ell(\boldsymbol{\theta} \mid \mathbf{y})$ is the likelihood, $p(\boldsymbol{\theta})$ is the prior, and $1 = \nu_1 > \cdots > \nu_d > 0$ are called the inverse temperatures. Note that the first chain has the same target distribution as the conventional RJ-MCMC algorithm does. We choose the inverse temperatures from the sigmoidal temperature ladder used in Gramacy and Taddy (2010) and Payne et al. (2020). Every a certain number of iterations (which is called a swap interval), all chains swap their parameters $\boldsymbol{\theta}$ with their neighboring chains with some probabilities. For a swap attempt between the jth and the $(j+1)$th chains, the acceptance probability is given by

$$\alpha_2 = \min\left\{1, \frac{p_j(\boldsymbol{\theta}_{j-1} \mid \mathbf{y}) \cdot p_{j-1}(\boldsymbol{\theta}_j \mid \mathbf{y})}{p_j(\boldsymbol{\theta}_j \mid \mathbf{y}) \cdot p_{j-1}(\boldsymbol{\theta}_{j-1} \mid \mathbf{y})}\right\},$$

where $\boldsymbol{\theta}_j$ is the parameter in the jth chain. The draws from the first chain are the MCMC samples from the desired posterior distribution. Generally, a chain with lower inverse temperature has higher acceptance rates in reversible jump moves, allowing it to reach regions that are hard to visit by chains with higher inverse temperatures. Samples from these regions can then be passed to chains with higher inverse temperatures by the swap procedure, which speeds up the exploration of the posterior sample space.

2.3.5 Selection of c

The hyperparameter c has profound effect on the asymptotic behavior of posterior distributions and thus it is rather important to carefully specify the order of c with respect to the sample size n. Following Assumption (C4), we set $1 - c = n^{-\alpha}$ so that the posterior consistency result in Theorem 2.3 can be guaranteed. In practice the constant α is unknown and the selection of c boils down to choosing appropriate positive α.

We propose to use Watanabe-Akaike information criterion (WAIC; Watanabe, 2010) to select α, which takes the form

$$\text{WAIC} = -2\sum_{i=1}^{n} \log\left(\frac{1}{S}\sum_{s=1}^{S} \ell(\boldsymbol{\theta}^s | y_i)\right) + 2p_{\text{WAIC}},$$

where y_i is a shorthand for $y(\mathbf{s}_i)$, $\boldsymbol{\theta}^s$ is the sth ($s = 1, \ldots, S$) MCMC sample of the parameters, and p_{WAIC} is a term quantifying model complexity. In addition to the widely used complexity term

$$p_{\text{WAIC}_1} = 2\sum_{i=1}^{n}\left\{\log\left(\frac{1}{S}\sum_{s=1}^{S}\ell(\boldsymbol{\theta}^s|y_i)\right) - \frac{1}{S}\sum_{s=1}^{S}\log\ell(\boldsymbol{\theta}^s|y_i)\right\},$$

a numerically more stable alternative

$$p_{\text{WAIC}_2} = V_{s=1}^{S}\log\ell(\boldsymbol{\theta}^s|y_i),$$

where $V_{s=1}^{S}$ represents the unbiased sample variance, is also recommended (Gelman et al., 2014). An α that leads to lower WAIC is preferred. Note that WAIC is applicable because our model assumes conditional independence of $\mathbf{y}$ given the parameters and the spatial dependence is modelled via the latent partition structure of the parameters.

2.4 Extensions to Other Hierarchical Models

The preceding Bayesian spanning tree partitioning prior model can be extended to other hierarchical model settings. Let $\{y_i,\ i = 1,\ldots,n\}$ be the observations at each vertex of an undirected graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$, where $\mathcal{G}_0$ encodes prior knowledge on the relationships among vertices to encourage neighboring vertices sharing identical models. Examples of such graphs can go beyond spatial domains to more complex domains such as brain networks, road networks or social networks.

Given a partition $\pi = \{\mathcal{C}_1,\ldots,\mathcal{C}_k\}$ of the vertices, we let $\mathbf{y}_{c_1},\cdots,\mathbf{y}_{c_k}$ denote the corresponding partition of observations. Conditional on the vector of latent cluster-specific model parameters, denoted as $\boldsymbol{\theta}_{(j)}$, $j = 1,\ldots,k$, and the vector of global model parameters $\boldsymbol{\eta}$, we assume a conditionally independent data-level model for $\mathbf{y}_{c_1},\cdots,\mathbf{y}_{c_k}$ as follows

$$\prod_{j=1}^{k} f(\mathbf{y}_{c_j} \mid \boldsymbol{\theta}_{(j)}, \boldsymbol{\eta}, \pi)$$

The Bayesian approach then proceeds by assigning prior models for $\boldsymbol{\theta}_j$ and $\boldsymbol{\eta}$ conditional on the graph partition π. Finally, the Bayesian spanning tree partitioning prior model introduced in Section 2.3.1 is adopted to model π.

There are many general settings in which the above hierarchical model with clustered latent variables arises as the data-level model can take various forms. One example is to consider generalized linear models (GLMs) for non-Gaussian data, which were also considered in Teixeira et al. (2015, 2019) for a spatial Poisson count response data. Commonly used non-Gaussian data level models include: (i) binary response at locations, modeled using logit or probit regression, and (ii) count data at locations, modeled using Poisson regression. We model the link function of mean responses using a clustered varying coefficient model,

$$g(E(y_i)) = \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta}_{(j)}, \text{ for } i \in \mathcal{C}_j \tag{2.9}$$

The prior models for the partitions can be assigned in the same way as in Section 2.3.1. If one simplifies the model by assuming a single common unknown partition for the whole vector of regression coefficients, a prior model such as a multivariate normal can be assigned for each $\boldsymbol{\beta}_{(j)}$ independently. For this single partition case, in addition to our prior model, one may also consider the spanning tree partitioning prior proposed in Teixeira et al. (2015, 2019).

Another example is to consider a locally stationary Gaussian process model, in a similar spirit of the treed Gaussian process approach (Gramacy and Lee, 2008; Konomi et al., 2014). Conditional on the partition, data within each cluster is modeled as a stationary Gaussian process with latent cluster-specific covariance parameters $\boldsymbol{\phi}_j$ and a global nugget effect τ^2, that is,

$$y_i = \mu_j + \omega_i^{(j)} + \epsilon_i, \text{ for } i \in \mathcal{D}_j \tag{2.10}$$

where $\omega_i^{(j)}$ is modeled as a zero mean Gaussian process with covariance function $C(\cdot; \boldsymbol{\phi}_j)$, and $\mathcal{D}_j$ is a subregion in the input space such that the nearest observed location from any input point within $\mathcal{D}_j$ belongs to $\mathcal{C}_j$. Given a partition, $[\{\mu_j, \boldsymbol{\phi}_j\}, \tau^2]$ are assigned with prior models following the typical Bayesian stationary Gaussian process conventions (Banerjee et al., 2014).

The RJ-MCMC algorithm presented in Section 2.3.4 can be adapted to sample the partitions and other parameters of the above models from their posterior distributions

$$
\begin{aligned}
p\left[\{\pi, k, \mathbf{w}\}, \{\boldsymbol{\theta}_{(j)}\}_{j=1:k}, \boldsymbol{\eta} \mid \mathbf{y}\right] \propto& \\
&\left\{\prod_{j=1}^{k} f(\mathbf{y}_{c_j} \mid \boldsymbol{\theta}_{(j)}, \boldsymbol{\eta}, \pi)\right\} p(\{\boldsymbol{\theta}_{(j)}\}_{j=1:k} \mid \pi) p(\pi, k, \mathbf{w}) p(\boldsymbol{\eta}) \quad (2.11)
\end{aligned}
$$

We remark that, in the Gaussian regression model, we marginalize out local cluster-specific parameters when sampling partitions to speed up mixing. But in the general case, the collapsed likelihood function may not be achievable. Nevertheless, in the birth, death and change moves in the RJ-MCMC algorithm, the calculation of the likelihood ratio can still be simplified since it only involves a subset of data that have changes in cluster memberships. Data augmentation tricks such as Albert and Chib (1993) for probit models and Polson et al. (2013) for logistic regression can also be applied to derive MCMC algorithms.

2.5 Simulation Studies

2.5.1 Simulation Setup

In this section, we assess the performance of the BSCC method by some simulation studies. For the ease of comparison with SCC, we use the same simulation setting as in Li and Sang (2019). 1000 spatial locations are generated uniformly in a square domain $[0, 1] \times [0, 1]$. We generate responses at each location from a linear model with an intercept term and two covariates

$$
y(\mathbf{s}_i) = x_1(\mathbf{s}_i)\beta_1(\mathbf{s}_i) + x_2(\mathbf{s}_i)\beta_2(\mathbf{s}_i) + \beta_3(\mathbf{s}_i) + \epsilon(\mathbf{s}_i), \quad \epsilon(\mathbf{s}_i) \overset{i.i.d.}{\sim} \mathrm{N}(0, \sigma^2). \qquad (2.12)
$$

We set the true coefficients to be constant within each cluster, the true value of σ to be 0.1, and the numbers of clusters to be 4 for β_1, 5 for β_2 and 6 for β_3, respectively. We consider different clustering patterns for each coefficient, which are shown in Figure 2.2. In

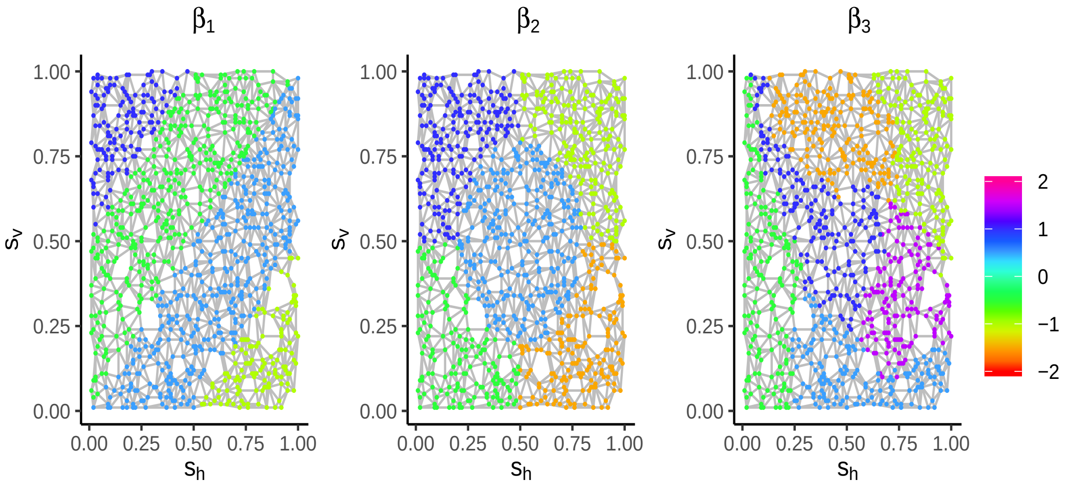

Figure 2.2: Spatial structures of true coefficients and the Delaunay triangulation used in BSCC.

particular, the shapes of true clusters for β_3 are designed to be highly irregular, with the goal of examining the capacity of the BSCC to capture irregular cluster boundaries.

The two covariates are generated such that there is a spatial correlation among locations. Since in practice many spatial covariates are correlated with each other, we also introduce linear dependence between $x_1(\mathbf{s}_i)$ and $x_2(\mathbf{s}_i)$. Specifically, let $\{\zeta_1(\mathbf{s}_i)\}$ and $\{\zeta_2(\mathbf{s}_i)\}$ be two independent realizations of a spatial Gaussian process with zero mean and an isotropic exponential covariance function given by $\operatorname{cov}\{\zeta_m(\mathbf{s}_i), \zeta_m(\mathbf{s}_j)\} = \exp\left(-\|\mathbf{s}_i - \mathbf{s}_j\|/\phi\right), m = 1, 2$, where ϕ is the range parameter controlling the strength of spatial correlation. Then $x_1(\mathbf{s}_i)$ and $x_2(\mathbf{s}_i)$ are obtained by a linear transformation given by $x_1(\mathbf{s}_i) = \zeta_1(\mathbf{s}_i), \quad x_2(\mathbf{s}_i) = r\zeta_1(\mathbf{s}_i) + \sqrt{1-r^2}\zeta_2(\mathbf{s}_i)$. We consider a moderate collinearity case by setting $r = 0.75$. For spatial correlation within each covariate, three cases are considered, namely, a weak correlation with $\phi = 0.1$, a moderate correlation with $\phi = 0.3$, and a strong correlation with $\phi = 1$. For each value of ϕ, the simulations are repeated 100 times with a same set of true values of coefficients.

We construct the initial graph using the Delaunay triangulation, removing edges longer than 0.1. We consider four candidates $\alpha = 0.0075, 0.0150, 0.1000, 0.3333$, which give $c =$

$0.05, 0.1, 0.5, 0.9$, respectively. The other hyperparameters are set to be $a_0 = b_0 = 1$ and $c_0 = d_0 = 10^{-6}$, and the standard deviation for the random walk proposal in the hyper step of our RJ-MCMC algorithm is chosen to be 0.9. For each simulated data set, we run $d = 8$ tempered chains in parallel with the lowest inverse temperature $t_d = 0.35$. We run each chain for $100,000$ iterations, discarding the first $50,000$. We set the thinning interval to be 20 iterations and the swap interval to be 100. A total of $2,500$ posterior samples are collected.

As is common in many Bayesian partition models (e.g., Denison et al., 1998; Gramacy and Lee, 2008; Payne et al., 2020), we use the maximum a posteriori (MAP) estimator for point estimation. The posterior distribution used here is the full $p\left[\boldsymbol{\beta}, \left\{\pi^{(m)}, k_m, \mathbf{w}^{(m)}\right\}_{m=1}^{p}, \sigma^2, \lambda \mid \mathbf{y}\right]$ derived from (2.5) (instead of the collapsed version in Equation 2.7). We also calculate the 95% highest posterior density (HPD) interval for each $\beta_m(\mathbf{s}_i)$ from the MCMC samples.

Most existing software for spatial clustering is designed for spatial response data or spatial points. The BSCC method is compared with the frequentist SCC method (Li and Sang, 2019) and a Dirichlet process mixture (DPM) model for spatial regressions proposed by Ma et al. (2020), due to the lack of other available software for multiple regressions with spatially clustered coefficients. In SCC a fixed MST is used and the tuning parameter for penalization is chosen by BIC. The original DPM model in Ma et al. (2020) includes a term for spatial random effects modeled by a Gaussian process. For fair comparison, we drop this term since our model doesn't include these smoothly varying effects (the results of the original version of DPM models are included in Appendix A.3.4). The DPM model is essentially a Bayesian linear varying coefficient model with a Dirichlet process prior on the coefficients to capture cluster patterns. Inference of the DPM model is based on MCMC, and we run the chain for $20,000$ iterations, discard the first half, and collect posterior samples every 10 iterations from the second half. MAP estimators are also used for the DPM model.

	$\alpha = 0.0075$	$\alpha = 0.0150$	$\alpha = 0.1000$	$\alpha = 0.3333$
	$(c = 0.05)$	$(c = 0.10)$	$(c = 0.50)$	$(c = 0.90)$
WAIC_1	49	37	13	1
WAIC_2	53	38	8	1

Table 2.1: Number of data sets (out of 100) with moderate spatial correlation in which WAIC prefers a certain value of α.

The performance of coefficient estimation is quantified by the mean squared error (MSE)

$$MSE_\beta = \frac{1}{np} \sum_{i=1}^{n} \sum_{m=1}^{p} \{\hat{\beta}_m(\mathbf{s}_i) - \beta_m(\mathbf{s}_i)\}^2.$$

We assess the performance of partition recovery by the Rand index, which is the proportion of agreements of the estimated partitions and the true ones. A Rand index that is closer to 1 indicates a better recovery of the true partition.

We implement the BSCC method in `R` using the `deldir` package for the Delaunay triangulation, the `igraph` package for graph operations, and the `ramcmc` package for the Cholesky update/downdate. The code will be made publicly available upon publication. The implementation of the SCC method is adapted from the `R` package `glmnet`. The DPM model is implemented in `R` using the `nimble` code provided in Ma et al. (2020). All computations were performed on a Linux server with two 2.4GHz 14-core processors and 64GB of memory.

2.5.2 Simulation Results

We first consider selecting the hyperparameter α (or equivalently, c) using WAIC. Table 2.1 shows the number of data sets with moderate spatial correlation in which WAIC prefers each candidate value of α. The value $\alpha = 0.0075$, which leads to $c = 0.05$, is preferred in most of the data sets by both criteria. As a result, the rest results of the simulation studies are all based on $c = 0.05$. The sensitivity analysis of α is shown in Appendix A.3.1.

We then assess the performance of BSCC based on 100 repeated experiments. The boxplots of MSEs of BSCC, SCC, and DPM under three different settings of spatial correlation

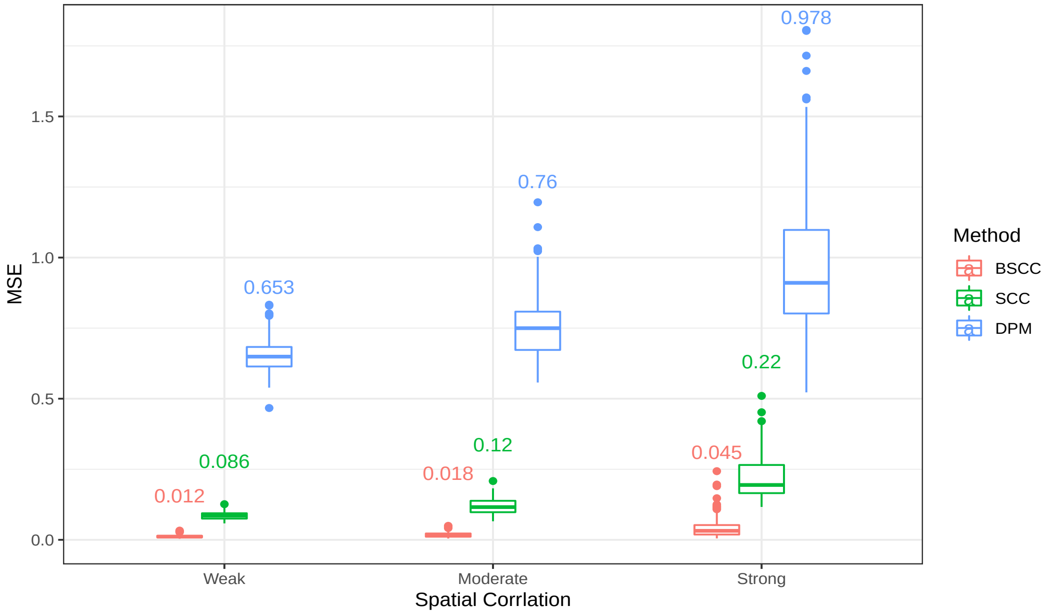

Figure 2.3: Boxplots of MSEs for BSCC, SCC, and DPM methods under 3 different settings of spatial correlation for predictors. 100 simulations are run for each setting. The average MSE_β over 100 simulations is shown above each box.

for predictors are shown in Figure 2.3. We can see that as the spatial correlation for predictors increases, all methods give higher MSEs. Under all settings, the MSE of BSCC is substantially lower than those of SCC and DPM. For instance, when the spatial range parameter of predictors is $\phi = 0.3$ (moderate correlation), the average MSE of BSCC is nearly 1/6 and 1/35 of the counterparts of SCC and DPM, respectively. Even when the spatial correlation is strong ($\phi = 1$), a less favorable case for parameter estimation, BSCC still provides a much more accurate coefficient estimation than SCC and DPM.

In terms of the performance in partition recovery, we compare the average Rand indices of BSCC, SCC, and DPM, over 100 simulations under each setting of spatial correlation. The results are presented in Table 2.2. BSCC considerably outperforms SCC and DPM in estimating the cluster patterns. Under weak or moderate spatial correlation, BSCC almost perfectly recovers the true partition, suggested by the high Rand indices close to 1. When the covariates are strongly correlated over the spatial domain, the Rand index of BSCC degenerates slightly, but overall still indicates remarkably accurate partition recovery.

	Rand index								
	β_1			β_2			β_3		
Spatial correlation	BSCC	SCC	DPM	BSCC	SCC	DPM	BSCC	SCC	DPM
Weak	0.986	0.716	0.686	0.990	0.819	0.781	0.997	0.852	0.822
Moderate	0.983	0.722	0.681	0.987	0.825	0.773	0.994	0.853	0.812
Strong	0.964	0.726	0.680	0.972	0.830	0.770	0.970	0.849	0.809

Table 2.2: The average Rand indices for BSCC, SCC, and DPM methods over 100 simulations.

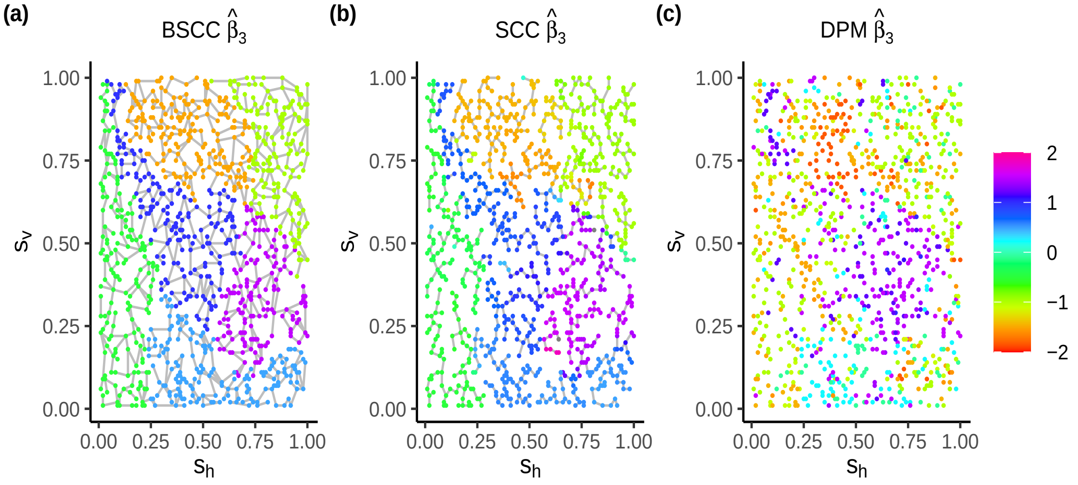

Figure 2.4: The estimated $\hat{\beta}_3(\mathbf{s}_i)$ from (a) BSCC, (b) SCC, and (c) DPM in one simulated data set with moderate spatially correlated predictors ($\phi = 0.3$). The MAP estimate of the spanning tree is shown in (a), and the minimum spanning tree used by SCC is shown in (b). Points with absolute values greater than 2 are marked in gray.

Next, we analyze the result from one simulated data set under the setting with a moderate spatial correlation ($\phi = 0.3$) in covariates. The data set that has a median MSE among 100 data sets is chosen for illustration.

Figure 2.4 shows the estimated $\hat{\beta}_3(\mathbf{s}_i)$ from BSCC, SCC, and DPM. While all methods can approximately capture the true patterns shown in Figure 2.2(c), BSCC gives a much more consistent result in terms of both partition recovery and parameter estimation. In contrast, the result from SCC has more mis-classified points and gives larger estimation errors. The result from DPM is noisier, and the clusters it identifies are not spatially contiguous. The results for $\hat{\beta}_1(\mathbf{s}_i)$ and $\hat{\beta}_2(\mathbf{s}_i)$ are similar and thus omitted. The numbers of clusters given by BSCC are 5 for β_1, 5 for β_2, and 6 for β_3, while the ones given by SCC are 92, 69, and 132, respectively. DPM results in 23 clusters for each coefficient. The results suggest that BSCC can recover the true partitions in a highly accurate way, including the irregularly shaped partition of β_3.

The improvement of BSCC over SCC is largely attributed to the fact that BSCC allows the spanning tree to be updated so that it has a consistent ordering with the true partitions. To illustrate, we show an example in Figure 2.5, which is a zoom-in version of Figure 2.2(c) and Figure 2.4(a, b) on the selected window $[0.6, 0.8] \times [0.65, 0.9]$. The points within the red circles are mis-classified by SCC but correctly classified by BSCC. The reason is that the MST in Panel (c) used in SCC is not able to induce the true partition; the mis-classified points are only connected to the neighboring cluster (marked by green points) instead of the true cluster (marked by orange ones), as they should be. As a result, there is no hope for SCC to recover the true partition due to the use of an inconsistent fixed ordering spanning tree. In contrast, the MCMC procedure in BSCC can fix this issue by updating the spanning tree such that it connects points in a more desirable way, as is shown in Figure 2.5(b).

Another advantage of BSCC over SCC is that the Bayesian inference procedure naturally comes with an uncertainty measure. Distributions of posterior samples of β_2 at four representative locations are shown in Figure 2.6, where 95% HPD intervals are marked by red segments. For a location in the interior of a cluster (i.e., far away from the true boundaries), which is shown in Panel (a), the posterior distribution is unimodal, and the HPD interval is narrow and covers its true coefficient (marked by the blue dashed line). The parameter estimation is accurate in this situation. Panels (b - d) show locations close to a true boundary of β_2. The posterior distribution in Panel (b) displays a similar pattern as Panel (a). A different pattern is shown in Panels (c) and (d), where the distributions are multimodal and have wider HPD intervals. Notice that lower modes in Panels (c) and (d) appear near the true values of β_2 in the neighboring clusters (indicated by the green dash-dotted line), and

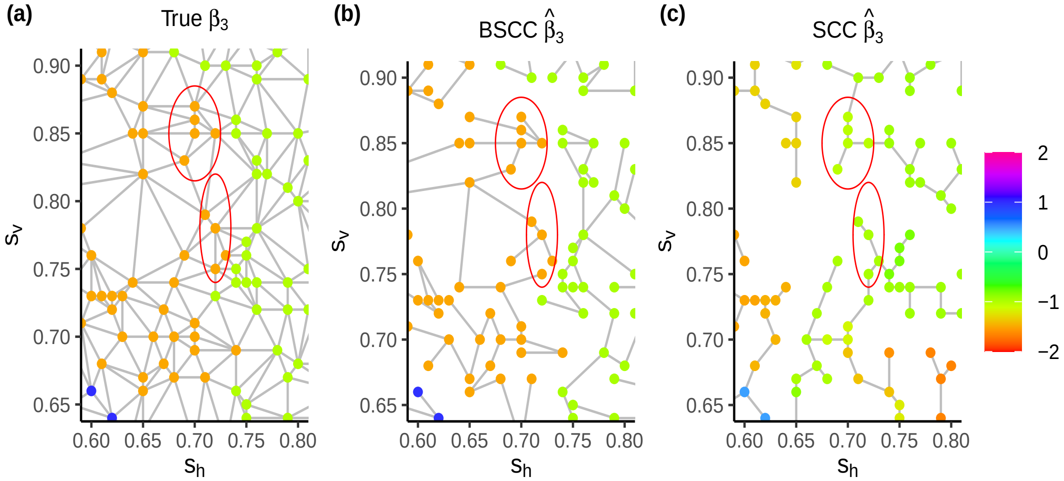

Figure 2.5: Zoomed version of Figure 2.2(c) and Figure 2.4(a, b) into the region $[0.6, 0.8] \times [0.65, 0.9]$. Some of the points mis-classified by SCC but correctly classified by BSCC are marked by red circles.

the HDP intervals also contain these values. In Panel (d) there is also a third mode between -0.5 and 0, probably because this location is assigned to some small-sized clusters in some of the MCMC samples. Overall, the posterior distributions assign a substantial amount of mass around the true coefficients. The multimodality reflects the uncertainty that a point near a boundary may be classified into either cluster around it. Posterior distributions of other locations display similar patterns.

Finally, we remark that the computational expense of BSCC is in general reasonable, thanks to the use of multiple computation strategies carefully designed for the collapsed RJ-MCMC algorithm in Section 2.3.4. With a moderate spatial correlation for covariates, the average time over 100 simulations to run 100, 000 iterations with 8 parallel chains is 20 minutes. As a comparison, DPM takes 56.3 minutes to finish 20, 000 MCMC iterations on average. Increasing spatial correlation has no impact on the running time.

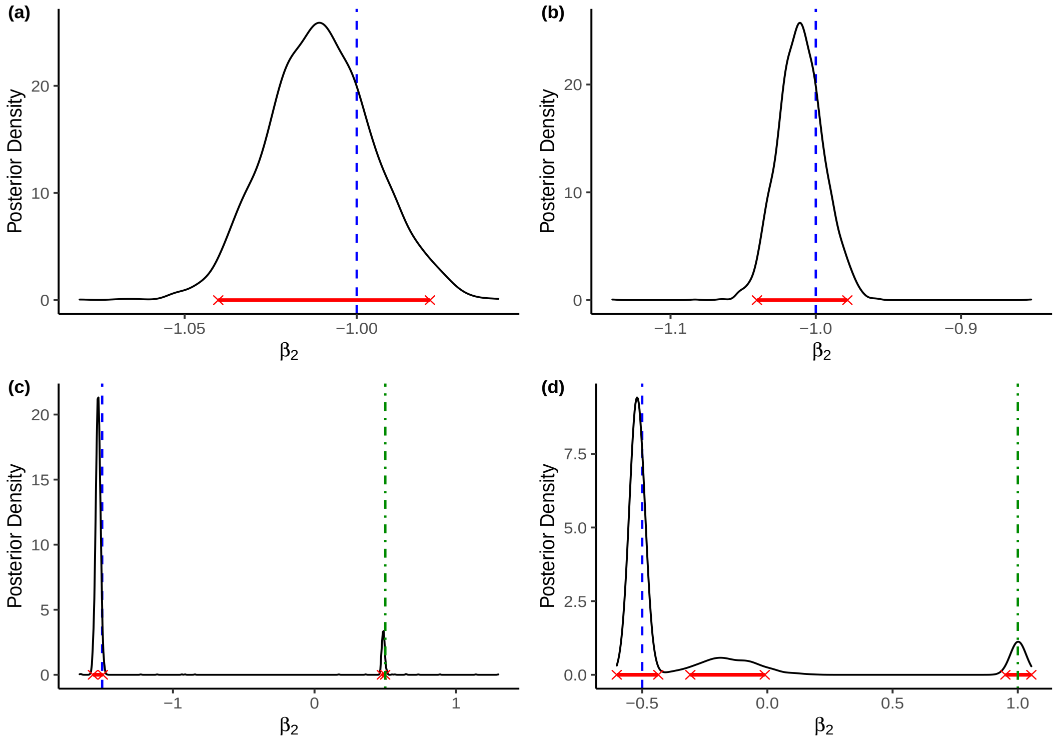

Figure 2.6: Distributions of posterior samples of β_2 at four locations (see the text for details). Red segments indicate 95% HPD intervals. True coefficient values are marked by blue dashed lines and true values of β_2 in neighboring clusters are marked by green dash-dotted lines. Note the scales of horizontal axes are different.

2.6 Real Data Analysis

2.6.1 Data Set

We apply our BSCC method to analyze the temperature-salinity (T-S) relationship of seawater in the Atlantic Ocean. Our goal is to identify the Antarctic Intermediate Water (AAIW) characterized by a negative T-S relationship (Talley, 2011). The identification of the AAIW could provide valuable information about Earth's climate change and thus is an important research question in geoscience. It is known that the T-S relationship is relatively homogeneous within certain regions but could change abruptly across the borders of individual water masses. Therefore, the T-S relationship is often assumed to be a spatially piecewise constant in oceanography.

The data of temperature and salinity is downloaded from National Oceanographic Data Center (`https://www.nodc.noaa.gov/OC5/woa13/`). We chose a random sample of $5,130$ spatial locations from the observations in the segment of the Atlantic basin along 25°W between 60°S and the equator. The distributions of both temperature and salinity have strong anisotropic spatial patterns as a result of the Ocean's geometry, which has a width of around 20,000 km and a thickness of about 4 km. To eliminate the anisotropy, we follow a rescaling method commonly used in oceanic studies (Vallis, 2017) by letting $(s_h, s_v) = (s_h^0/L, s_v^0/H)$, where s_h^0 (s_v^0) is the original latitude (depth) and L (H) is the horizontal (vertical) length of the ocean.

2.6.2 Analysis Results

The relationship of temperature and salinity is modeled by

$$\mathit{Sal}(\mathbf{s}_i) = \beta_0(\mathbf{s}_i) + \beta_1(\mathbf{s}_i)\mathit{Temp}(\mathbf{s}_i) + \epsilon(\mathbf{s}_i),$$

where $\mathit{Sal}(\mathbf{s}_i)$ and $\mathit{Temp}(\mathbf{s}_i)$ are the salinity and temperature at location $\mathbf{s}_i = (s_{h,i}, s_{v,i})$, respectively, $\beta_0(\mathbf{s}_i)$ is the intercept, and $\beta_1(\mathbf{s}_i)$ denotes the T-S relationship of interest. Both

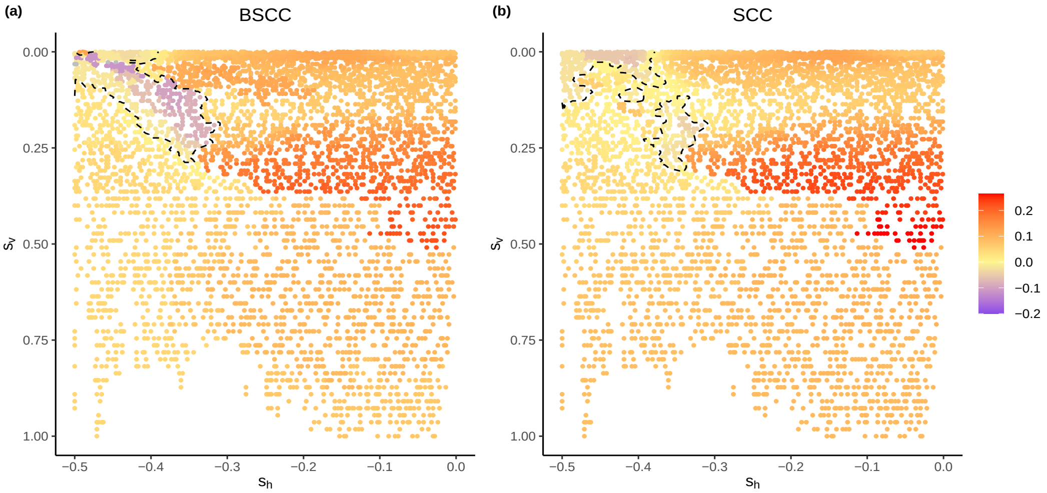

Figure 2.7: The T-S relationship β_1 estimated from (a) BSCC and (b) SCC. The contour of $\beta_1 = 0$ given by interpolation is shown as the black dashed line.

β_1 and β_0 are assumed to be spatially piecewise constant. We adopt the same prior as the simulation studies described in Section 2.5.1 except that we only consider a candidate set of $\alpha \in \{0.0075, 0.015, 0.1\}$ due to computational expense. The optimal model selected by WAIC corresponds to $\alpha = 0.1$, which gives $c = 0.574$. We run $d = 20$ chains with lowest temperature $t_d = 0.1$. Each chain is run for $1,500,000$ iterations with the first $1,000,000$ as burn-in period. The swap and the thinning intervals are set to be 100 and 50, respectively, giving $10,000$ posterior samples in total. Typically such a long chain is needed for large data sets in Bayesian high dimensional regression models to get reliable uncertainty estimates (e.g., Zhou and Guan, 2019; Guan and Stephens, 2011).

The traceplot of posterior samples of σ^2 displays satisfactory convergence and mixing performance. The slope estimates from BSCC as well as SCC are shown in Figure 2.7, and the estimated boundaries of the AAIW regions (points with negative slope estimates) are marked by black dashed lines in Figure 2.7. BSCC gives 68 clusters for the slope β_1. In contrast, SCC gives 1141, which is too large for interpretation. A band-shaped AAIW region located near the sea surface from $s_h = -0.30$ to $s_h = -0.50$ is identified by BSCC. Its encompassing region covers the well-recognized generation site of AAIW and the low-salinity

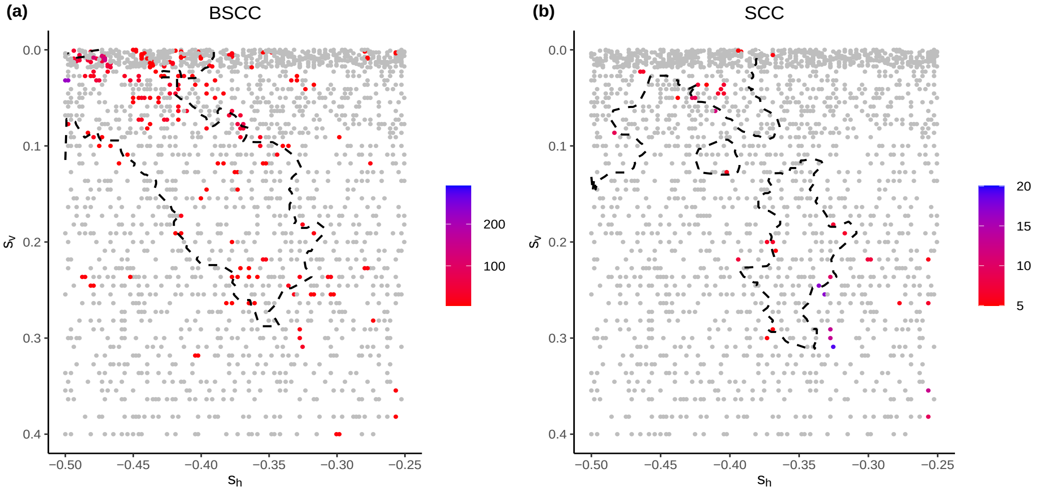

Figure 2.8: The magnitude of spatial difference quotient of the T-S relationship estimated by (a) BSCC and (b) SCC. Note the color scales in two panels are different. Points with magnitudes less than 5 are marked in gray.

tongue which is believed to be associated with AAIW (Talley, 2011). We also notice that BSCC gives a spatially contiguous region of AAIW, while SCC does not.

As suggested by geophysical theory, the T-S relationship may change dramatically across the boundary of AAIW (Talley, 2011). We quantify the change of the estimated T-S relationship by the magnitude of spatial difference quotient (Simmonds, 2012), which is given by

$$D(\mathbf{s}_i) = \left[\frac{\{\beta_1(\mathbf{s}_i)-\beta_1(\mathbf{s}_{i_1})\}^2}{d_1^2\sin^2\gamma} + \frac{\{\beta_1(\mathbf{s}_i)-\beta_1(\mathbf{s}_{i_2})\}^2}{d_2^2\sin^2\gamma} - \frac{2\{\beta_1(\mathbf{s}_i)-\beta_1(\mathbf{s}_{i_1})\}\{\beta_1(\mathbf{s}_i)-\beta_1(\mathbf{s}_{i_2})\}\cos\gamma}{d_1 d_2\sin^2\gamma}\right]^{\frac{1}{2}},$$

where $\mathbf{s}_{i_1}$ and $\mathbf{s}_{i_2}$ are two nearest location of $\mathbf{s}_i$, d_j is the distance between $\mathbf{s}_i$ and $\mathbf{s}_{i_j}$, $j = 1, 2$, and γ is the angle between vectors $(s_{h,i_1} - s_{h,i}, s_{v,i_1} - s_{v,i})$ and $(s_{h,i_2} - s_{h,i}, s_{v,i_2} - s_{v,i})$. Figure 2.8 shows the results from BSCC and SCC. Consistent with the theoretical results in geophysics, the change of β_1 given by BSCC is abrupt around the boundary. For the results from SCC, the change has much smaller magnitude, partly due to the shrinkage effect of the L_1 penalty on the differences between neighboring regression coefficients.

Finally we illustrate the uncertainty of the T-S relationship estimation in Figure 2.9. The

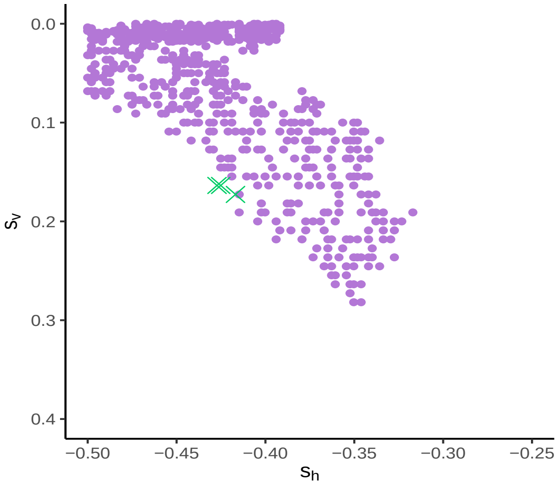

Figure 2.9: Potential AAIW regions estimated from BSCC. Points with negative $\hat{\beta}_1$ from MAP estimation are shown in purple. Locations where 95% HPD intervals of β_1 include 0 are marked by green crosses. Note that only the region $[-0.5, -0.25] \times [0, 0.4]$ is shown.

T-S relationship of purple points are estimated to be negative with high certainty. We find 3 locations along the boundary of the AAIW region whose 95% HPD intervals of β_1 include 0, and they can be viewed as part of the potential boundary of AAIW.

2.7 Conclusions and Discussion

In this chapter, a novel spatial regression method, called Bayesian Spatially Clustered Coefficient regression, is developed to estimate the clustered relationship among spatial variables. Our BSCC method is based on a model-based spatially contiguous clustering method defined via connected components of an undirected graph, which we prove can be induced by a spanning tree and a suitable subset of its edge set. A prior for spatial partitions is therefore developed hierarchically by assigning priors to spanning trees as well as their edge sets. We prove that the BSCC model achieves posterior consistency for point estimation under this prior. However, results for posterior selection consistency (i.e., the property that the posterior distribution of partitions concentrates at the true partition) are non-trivial to prove, and we leave this for future research.

For computation, we propose an RJ-MCMC algorithm to sample spanning trees and partitions from their posterior distributions. Various computation methods such as parallel tempering are utilized to facilitate convergence. Our simulation studies demonstrate that BSCC remarkably outperforms its competitors SCC and DPM. In particular, BSCC achieves nearly 80% reduction in MSE in our simulation studies when compared with its frequentist counterpart, SCC, partially for the reason that the MCMC procedure can effectively fix the mis-classification in SCC by proposing a more desired spanning tree. We also present an application of BSCC to the detection of water masses by estimating the spatial clustering patterns of T-S relationship in the Atlantic basin.

One potential research direction is to further improve the convergence and mixing of the BSCC algorithm. A long burn-in period is typically needed before the chain converges for our simulated and real data. For binary tree based methods, efficient proposals for new partitions have been well-studied in literature (Chipman et al., 1998, 2010; Wu et al., 2007). For the proposed spanning tree based model, we have tried to propose new partitions adaptively by splitting an existing cluster at boundaries. However, we did not observe substantial improvement in terms of mixing and convergence (see Appendix A.4.2 for details). Modifications of proposals in the current RJ-MCMC algorithm are currently under investigation. Nevertheless, we remark that based on our numerical experiments, even when the chain does not fully converge, one can often still get reasonably accurate point estimations of partitions and coefficient values, though the reliability of uncertainty measures such as HPD intervals and Bayesian model averaging might be a concern. Hyperparameter selection is another remaining challenge in the model. Despite the utility of the proposed hyperparameter selection method in Section 2.3.5, a careful choice of the candidate set for α is still required to achieve better performance when one has little information about the number of clusters *a priori*.

Our current model in (2.1) assumes that the intercept and other regression coefficients are spatially piecewise constant. It is straightforward to generalize (2.1) to be $y(\mathbf{s}_i) = \mathbf{x}_1(\mathbf{s}_i)^\mathsf{T}\boldsymbol{\beta}(\mathbf{s}_i)+\mathbf{x}_2(\mathbf{s}_i)^\mathsf{T}\boldsymbol{\alpha}(\mathbf{s}_i)+\epsilon(\mathbf{s}_i)$, where $\boldsymbol{\beta}$ has clustering patterns and $\boldsymbol{\alpha}$ is smoothly varying. Incorporating a spatial Gaussian process random effect into the BSCC model is a special case of it.

3. A NONSTATIONARY SOFT PARTITIONED GAUSSIAN PROCESS MODEL VIA RANDOM SPANNING TREES

3.1 Introduction

Gaussian processes (GPs) have been a widely used modeling tool in spatial statistics, machine learning, and computer experiments. In the past decades, nonstationary GPs have attracted much attention for their flexibility in modeling varying spatial dependence structures over the spatial domain. Despite many progress on nonstationary spatial process models in the literature, it is still considered one of the most important but challenging topics in spatial statistics to develop flexible, computationally efficient, and theoretically justified nonstationary models. The idea of locally stationary process models has gained great popularity in spatial statistics and machine learning literature, due to its advantages in adapting to local and nonstationary data features and naturally allowing for reduced computations using local model results. However, there are several vital questions surrounding such methods: i) how many partitions to use? ii) how to identify locally stationary partitions (regions)? iii) how to achieve consensus predictions from local models, especially at boundaries?

Some existing work on locally stationary GPs relies on predetermined subregions. Park et al. (2011) used uniform grids for roughly evenly distributed data points. For unevenly distributed data points, k-d tree partitions with rectangular shapes (Shen et al., 2006) and spatial hierarchical clustering algorithms (Heaton et al., 2017) were used in the literature. Gerber and Nychka (2021) considered an overlapping domain partitioning method and used a parallel cross-validation algorithm to estimate local covariance parameters and perform spatial predictions. Alternative to these are the model-based partitioning methods. Risser et al. (2019) considered GP models based on Gaussian mixture clustering of spatial locations, where GP parameters and partitions are estimated separately. Bolin et al. (2019) developed a mixture of GP model for data on a uniform grid, where the clusters are modeled by a Markov random field and hence may not be spatially contiguous. The binary-treed GP models proposed in Gramacy and Lee (2008) partitioned input space into non-overlapping regions by making binary splits recursively, and hence only producing rectangular shaped clusters with boundaries always parallel to the input-space axes. Kim et al. (2005) assumed the partition is defined by a number of centering locations such that points within a cluster are closer to its center than any other centers, which leads to convex-polygon-shaped clusters (a.k.a. Voronoi cells). The Vonoroi tessellation based method was extended in Pope et al. (2021) by allowing a subregion to be formed by multiple convex polygons, which does not guarantee spatial contiguity of subregions, and in Gosoniu and Vounatsou (2011) by assuming a mixture of cell-specified models with distance-based weights. Despite the benefits of locally stationary models, a common criticism of many methods is that they are lack of a coherent global process for inference and predictions. Moreover, the constraints imposed on the shape of clusters in the current literature considerably limit the applications and interpretabilities of local stationary models for real problems, where it is of interest for practitioners to detect and locate spatial nonstationarities that may have highly irregular structures. Most recently, the spanning-treed partitioning model has been demonstrated as an efficient modeling tool for highly flexible spatially contiguous cluster shapes (Li and Sang, 2019; Teixeira et al., 2019; Luo et al., 2021b). Nonetheless, these works have been restricted to the partition of a finite set of observed locations in regression settings. Besides the aforementioned locally stationary GP models, a variety of nonstationary covariance functions of GP have been proposed to model the heterogeneity of spatial dependence based on the ideas of kernel convolutions, dimension expansions, spatial deformations,basis representations and stochastic partial differential equations. We refer the interested readers to Risser (2016) and Fouedjio (2017) for a comprehensive review.

In light of these challenges and limitations, our contribution is to develop a new class of nonstationary GP models with flexible and desirable dependence structures for high-dimensional spatial values. The proposed nonstationary model is constructed from locally stationary stochastic processes on a partitioned domain. We propose a general framework to extend an arbitrary partition on a finite set of reference knots to the whole spatial domain, by introducing a soft space partition process that utilizes neighborhood information. Built upon the latent space partition, a valid global spatial process model, called a soft partitioned GP (SPGP), is further defined to knit together local models such that the predictive distributions admit Gaussian mixture structures that can lead to better performance in prediction and uncertainty quantification. The idea of building spatial processes from finite dimensional models has shown great promise in recent literature (see, e.g., Lindgren et al., 2011; Datta et al., 2016). Our formulation adds to this line of work, but the motivation and model specifications are vastly different from the existing literature.

To address the key and challenging issue of learning space partitions with flexible shapes and sizes, we embed the proposed SPGP model in a Bayesian hierarchical modeling setting and assign a spanning-treed partition prior on the finite reference set, although our general framework of constructing SPGP can adopt any other partition priors such as binary trees, Voronoi tessellations, and product partition models. We formally define spatially contiguous space partitions and prove that, a key and unique advantage of the proposed partition model over existing partition methods is that it fully accommodates all possible contiguous partitions. Local partitions can be automatically learned from the data for discontinuities/abrupt changes recovery, and smoothness in spatial random fields can be captured by Bayesian model averaging of SPGP. We also make a theoretical contribution to study the Bayesian posterior concentration concerning the infill asymptotic behavior of this Bayesian nonstationary process model. These theoretical results provide important guidance for hyper-parameter selections in practice. To the best of our knowledge, Bayesian theoretical properties of locally partitioned GP models haven't been investigated in the literature. Moreover, the modeling framework allows flexible choices of reference knots, which, if selected to be smaller sets, naturally deliver a speed-up computation algorithm. We offer several other computation strategies to take advantage tree structures and recently developed block-based fast algorithms for GP models with massive spatial data.

The rest of this chapter is organized as follows. Section 3.2 describes a general framework to construct a SPGP. Section 3.3 develops a SPGP model based on random spanning-treed partitions and states its theoretical properties. In Section 3.4, we discuss some computational strategies. We then demonstrate the model performance with synthetic data in Section 3.5 and with real precipitation data in Section 3.6. Finally, Section 3.7 concludes the chapter with some discussions. Technical proofs, details of posterior inference, and supplementary results on the synthetic and real data are provided in Supplementary Materials.

3.2 A Soft Partitioned Gaussian Process

In many environmental applications, spatial data often exhibits a dependence structure that is not homogeneous in space. Oftentimes data within a subregion are relatively homogeneous while there could be substantial difference among the subregions. To introduce a valid global process model to characterize spatially heterogeneous dependence, we begin with a finite partitioned Gaussian distribution model on $\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\} \subseteq \mathcal{D} \subseteq \mathbb{R}^d$, and extend the finite partition to a soft partition process to probabilistically model the cluster memberships of any given locations in Section 3.2.2. These two modeling components are then used to build the soft partitioned GP in Section 3.2.3.

3.2.1 Gaussian Densities on a Finite Partitioned Set

We consider the case where the set $\mathcal{S}$ is partitioned into a few disjoint subsets so that each subset can be treated as having homogeneous spatial dependence and modeled separately. Formally, for a generic set $\mathcal{A}$ (which can be either finite or infinite), we say $\pi_k(\mathcal{A}) = \{\mathcal{A}_1, \dots, \mathcal{A}_k\}$, where $\mathcal{A}_j \subseteq \mathcal{A}$ for $j = 1, \dots, k$, is a *partition* of $\mathcal{A}$ if it satisfies $\cup_{j=1}^{k} \mathcal{A}_j = \mathcal{A}$ and $\mathcal{A}_j \cap \mathcal{A}_{j'} = \emptyset$ for all $j \neq j'$, and each $\mathcal{A}_j$ is called a *cluster*. In spatial settings, it is desired to impose spatial contiguity constraints on partitions such that each cluster can be interpreted as a subregion. The modeling of $\pi_k(\mathcal{S})$ is a key ingredient in our method. We will introduce a new stochastic process based method via random spanning trees to model $\pi_k(\mathcal{S})$ with an unknown k in Section 3.3.1. But for now we assume $\pi_k(\mathcal{S})$ to be given, to stress the fact that the soft partitioned GP modeling is a general framework that can be built upon a wide range of partition models.

Given a generic partition $\pi_k(\mathcal{S})$, we allow each $\mathbf{w}(\mathcal{S}_j) = \{w(\mathbf{s}) : \mathbf{s} \in \mathcal{S}_j\}$ to be a realization of different zero-mean Gaussian processes characterized by a covariance function $C(\cdot, \cdot|\boldsymbol{\theta}_j)$, i.e.,

$$\mathbf{w}(\mathcal{S}_j)|\pi_k(\mathcal{S}) \sim \mathrm{N}_{n_j}\left\{\mathbf{0}, \mathbf{C}(\mathcal{S}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)\right\} \tag{3.1}$$

independently for all $j = 1, \dots, k$, where $n_j = |\mathcal{S}_j|$ is the number of locations in the cluster $\mathcal{S}_j$. The joint distribution of $\mathbf{w}(\mathcal{S}) = \{\mathbf{w}(\mathcal{S}_1), \dots, \mathbf{w}(\mathcal{S}_k)\}$ conditional on $\pi_k(\mathcal{S})$ is therefore Gaussian with a block-diagonal covariance matrix whose jth block is $\mathbf{C}(\mathcal{S}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)$. In Section 3.2.3, we will extend this joint distribution to a well-defined stochastic process whose covariance function locally admits the form $C(\cdot, \cdot|\boldsymbol{\theta}_j)$ in the interior of a cluster.

3.2.2 A Soft Partition Process

In order to extend the Gaussian density on a finite partitioned set to a valid process, a key fist step is to extend the partition on a finite set $\mathcal{S}$ to a partition process on $\mathcal{D}$. Given $\pi_k(\mathcal{S})$, we define $z(\mathbf{s}) \in \{1, \dots, k\}$ be the cluster membership of any location $\mathbf{s} \in \mathcal{D}$ such that $z(\mathbf{s}) = j$ with probability one if $\mathbf{s} \in \mathcal{S}_j$.

Let $\mathcal{U}$ be any finite subset of $\mathcal{D}$ such that $\mathcal{U} \cap \mathcal{S} = \emptyset$. Let $N_{\mathbf{u},\ell}$ be the ℓth nearest neighbor of $\mathbf{u} \in \mathcal{U}$ in $\mathcal{S}$ under a given metric $d(\cdot, \cdot)$ on $\mathcal{D}$. Intuitively, a location $\mathbf{u} \in \mathcal{U}$ is expected to share the same cluster membership as one of its neighbors in $\mathcal{S}$, and if $\mathbf{u}$ is near the boundary of a cluster in $\pi_k(\mathcal{S})$, such that $z(N_{\mathbf{u},\ell}) \neq z(N_{\mathbf{u},\ell'})$ for some small $\ell \neq \ell'$, it is more ideal to assign a cluster membership to $\mathbf{u}$ probabilistically to reflect the partitioning uncertainty and subsequently allow for model averaging. This motivates us to consider a random membership assignment model for $\mathbf{z}(\mathcal{U})$ following a similar spirit as the soft decision boundary proposed by Linero and Yang (2018) in a regression additive decision tree setting. More specifically, we assume that $\mathbf{z}(\mathcal{U})$ is independent from $\mathbf{z}(\mathcal{S})$ and each $z(\mathbf{u})$, $\mathbf{u} \in \mathcal{U}$, follows an independent and identical (iid) categorical distribution with probabilities $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_L)$ that sum to 1:

$$\mathbb{P}\left\{z(\mathbf{u}) = z(N_{\mathbf{u},\ell})\right\} = \alpha_\ell, \quad \text{for } \ell = 1, \ldots, L, \tag{3.2}$$

where L is the pre-specified maximal number of neighbors to be considered. We denote this catergorical distribution by $\text{Cat}\left\{z(N_{\mathbf{u},1}), \ldots, z(N_{\mathbf{u},L}) | \alpha_1, \ldots, \alpha_L\right\}$. This assumption essentially means that $\mathbf{u}$ is assigned the same cluster membership as its ℓth nearest neighbor in $\mathcal{S}$ with probability α_ℓ, for $\ell = 1, \ldots, L$. The probabilities $\boldsymbol{\alpha}$ are assumed to be known *a priori*. Choices of $\boldsymbol{\alpha}$ include (i) $\alpha_\ell = 1/L$ for all ℓ, which leads to a discrete uniform distribution on the neighboring clusters, and (ii) distance-based probabilities such as $\alpha_\ell \propto 1/d(\mathbf{u}, N_{\mathbf{u},\ell})$, for $\ell = 1, \ldots, L$.

It is easy to see that this construction defines a stochastic process $\{z(\mathbf{v}) : \mathbf{v} \in \mathcal{D}\}$ given $\pi_k(\mathcal{S})$ that takes value in $\{1, \ldots, k\}$, such that the joint distribution for any finite set $\mathcal{V} \subseteq \mathcal{D}$ satisfies $p\big(\mathbf{z}(\mathcal{V})\big) = \prod_{\mathbf{v} \in \mathcal{V}} p\big(z(\mathbf{v})\big)$, where $p\big(z(\mathbf{v})\big)$ is a degenerated distribution on j if $\mathbf{v} \in \mathcal{S}_j$ or the categorical distribution in (3.2) if $\mathbf{v} \notin \mathcal{S}$. We refer to this process as an L nearest neighbor *soft partition process* (L-SPP) conditional on $\pi_k(\mathcal{S})$. A realization of it at $\mathcal{V}$ in fact defines a partition $\pi_k(\mathcal{V}) = \{\mathcal{V}_1, \ldots, \mathcal{V}_k\}$, where $\mathcal{V}_j = \{\mathbf{v} \in \mathcal{V} : z(\mathbf{v}) = j\}$ for $j = 1, \ldots, k$.

3.2.3 Extension to a Soft Partitioned Gaussian Process

To extend (3.1) to a legitimate spatial process on $\mathcal{D}$, we first define the distribution of $\mathbf{w}(\mathcal{U})$ given $\mathbf{w}(\mathcal{S})$ for any finite set $\mathcal{U}$ that is disjoint from $\mathcal{S}$. Given a realization of the L-SPP $\mathbf{z}(\mathcal{U}) = \big(z(\mathbf{u}_1), \ldots, z(\mathbf{u}_r)\big)$ conditional on $\pi_k(\mathcal{S})$, $\mathcal{U}$ can be partitioned into clusters $\mathcal{U}_j = \{\mathbf{u} \in \mathcal{U} : z(\mathbf{u}) = j\}$. The conditional distribution of $\mathbf{w}(\mathcal{U})$ given $\mathbf{w}(\mathcal{S}_j)$, $\mathbf{z}(\mathcal{U}_j)$, and $\pi_k(\mathcal{S})$ is assumed to be

$$\mathbf{w}(\mathcal{U}_j) | \mathbf{w}(\mathcal{S}_j), \mathbf{z}(\mathcal{U}_j), \pi_k(\mathcal{S}) \sim \text{N}_{r_j}\left\{\boldsymbol{\mu}(\mathcal{U}_j | \mathcal{S}_j, \boldsymbol{\theta}_j),\ \boldsymbol{\Sigma}(\mathcal{U}_j | \mathcal{S}_j, \boldsymbol{\theta}_j)\right\}, \tag{3.3}$$

independently for $j = 1, \ldots, k$, where $r_j = |\mathcal{U}_j|$, and

$$\boldsymbol{\mu}(\mathcal{U}_j|\mathcal{S}_j, \boldsymbol{\theta}_j) = \mathbf{C}(\mathcal{U}_j, \mathcal{S}|\boldsymbol{\theta}_j)\mathbf{C}^{-1}(\mathcal{S}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)\mathbf{w}(\mathcal{S}_j), \tag{3.4}$$

$$\boldsymbol{\Sigma}(\mathcal{U}_j|\mathcal{S}_j, \boldsymbol{\theta}_j) = \mathbf{C}(\mathcal{U}_j,\mathcal{U}_j|\boldsymbol{\theta}_j) - \mathbf{C}(\mathcal{U}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)\mathbf{C}^{-1}(\mathcal{S}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)\mathbf{C}(\mathcal{S}_j,\mathcal{U}_j|\boldsymbol{\theta}_j). \tag{3.5}$$

Combining (3.1) and (3.3), for *any* finite subset $\mathcal{V}$ of $\mathcal{D}$ with the associated cluster memberships $\mathbf{z}(\mathcal{V})$, the density of $\mathbf{w}(\mathcal{V})$ given $\mathbf{z}(\mathcal{V})$ and $\pi_k(\mathcal{S})$ is given by

$$p\big(\mathbf{w}(\mathcal{V})|\mathbf{z}(\mathcal{V})\big) = \int p\big(\mathbf{w}(\mathcal{U})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{U})\big)\, p(\mathbf{w}(\mathcal{S})) \prod_{\{\mathbf{s}\in\mathcal{S}\setminus\mathcal{V}\}} \mathrm{d}(\mathbf{w}(\mathbf{s})) \quad \text{where } \mathcal{U} = \mathcal{V} \setminus \mathcal{S}. \tag{3.6}$$

The dependence on $\pi_k(\mathcal{S})$ and parameters $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)$ is made implicit in (3.6) for conciseness. Note that if $\mathcal{V} \subseteq \mathcal{S}$ then $p(\mathbf{w}(\mathcal{U})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{U})) = 1$ and if $\mathcal{S} \setminus \mathcal{V} = \emptyset$ then the integration in (3.6) is not needed. A mean-zero GP on $\mathcal{D}$ is therefore defined by (3.6) conditional on an L-SPP on $\mathcal{D}$, with a covariance function

$$C^{\ddagger}(\mathbf{v}, \mathbf{v}'|z(\mathbf{v}), z(\mathbf{v}'), \boldsymbol{\Theta}) = \begin{cases} C\big(\mathbf{v}, \mathbf{v}'|\boldsymbol{\theta}_j\big), & \text{if } \mathbf{v}, \mathbf{v}' \in \mathcal{D}_j \text{ for some } j \in \{1, \ldots, k\}, \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{D}_j = \{\mathbf{s} \in \mathcal{D} : z(\mathbf{s}) = j\}$ is the collection of all locations in $\mathcal{D}$ that are assigned to the jth cluster.

Marginalizing out the L-SPP, the unconditional density of $p(\mathbf{w}(\mathcal{V}))$ for *any* finite subset $\mathcal{V} \subseteq \mathcal{D}$ is therefore given by a Gaussian mixture

$$p\big(\mathbf{w}(\mathcal{V})\big) = \sum_{\mathbf{z}(\mathcal{V})} p\big(\mathbf{w}(\mathcal{V})|\mathbf{z}(\mathcal{V})\big)p\big(\mathbf{z}(\mathcal{V})\big), \tag{3.7}$$

where the summation is over all possible combinations of cluster memberships $\mathbf{z}(\mathcal{V})$. As shown in Appendix B.1.1, the density (3.7) satisfies Kolmogorov's consistency criteria and thus implies a valid spatial process on $\mathcal{D}$, which we call an L nearest neighbor *soft partitioned*

Gaussian process (L-SPGP) conditional on a partition $\pi_k(\mathcal{S})$. The covariance function of this process is given by

$$C^{\ddagger}(\mathbf{v}, \mathbf{v}'|\boldsymbol{\Theta}) = \sum_{j=1}^{k} \kappa_j C(\mathbf{v}, \mathbf{v}'|\theta_j), \tag{3.8}$$

where the weights $\kappa_j = \mathbb{P}\{z(\mathbf{v}) = j\} \times \mathbb{P}\{z(\mathbf{v}') = j\}$ is the probability that both locations belong to the jth cluster. Note that for any location $\mathbf{v} \in \mathcal{D}$, $\mathbb{P}\{z(\mathbf{v}) = j\} = \sum_{\ell=1}^{L} \alpha_j \mathbb{1}(N_{\mathbf{v},\ell} \in \mathcal{S}_j)$ if $\mathbf{v} \notin \mathcal{S}$, and $\mathbb{P}\{z(\mathbf{v}) = j\} = 1$ if $\mathbf{v} \in \mathcal{S}_j$, where $\mathbb{1}(\cdot)$ is an indicator function. Therefore, κ_j is completely determined by the neighborhood structures. In particular, $C^{\ddagger}(\mathbf{v}, \mathbf{v}'|\Theta)$ reduces to $C(\mathbf{v}, \mathbf{v}'|\theta_j)$ if $\mathbf{v}, \mathbf{v}' \in \text{Int}(\mathcal{S}_j)$, where $\text{Int}(\mathcal{S}_j)$ is the interior space corresponding to cluster $\mathcal{S}_j$ defined by $\text{Int}(\mathcal{S}_j) := \mathcal{S}_j \cup \{\mathbf{u} \in \mathcal{D}\backslash\mathcal{S} : \text{all of the } L \text{ nearest neighbors of } \mathbf{u} \text{ in } \mathcal{S} \text{ belong to } \mathcal{S}_j\}$. If $C(\mathbf{v}, \mathbf{v}'|\theta_j)$ is taken to be a stationary covariance function, then L-SPGP is locally stationary within $\text{Int}(\mathcal{S}_j)$. Note that this process can also be viewed as a finite *mixture* of GPs defined on $\mathcal{D}$ with spatially varying mixture weights, where each mixture component is $\text{GP}(0, C)$. Although the covariance function (3.8) seems similar to the ones obtained from discrete kernel convolution, they are essentially different approaches, in the the sense that the weights $\{\kappa_j\}$ comes from the uncertainty of partitioning a domain instead of some kernel functions.

For further illustration of L-SPGP, let us consider two examples.

Example 3.1. Let $\mathcal{S}$ be the locations where the realization of the process $\{w(\mathbf{v})\}$ is observed and $\mathbf{u} \notin \mathcal{S}$ be a location on which we want to do prediction. The conditional (also called predictive or kriging) distribution is given by a Gaussian mixture

$$w(\mathbf{u})|\mathbf{w}(\mathcal{S}), \pi_k(\mathcal{S}) \sim \sum_{\ell=1}^{L} \alpha_\ell \ \text{N}_1 \left(\boldsymbol{\mu}(\mathbf{u}|\mathcal{S}_{j(\ell)}, \boldsymbol{\theta}_{j(\ell)}), \boldsymbol{\Sigma}(\mathbf{u}|\mathcal{S}_{j(\ell)}, \boldsymbol{\theta}_{j(\ell)})\right), \tag{3.9}$$

where $j(\ell) = z(N_{\mathbf{u},\ell})$. The uncertainty of cluster memberships of $\mathbf{u}$ for prediction is captured by the Gaussian mixture structure. We refer the mean and variance of the Gaussian mixture in (3.9) as kriging mean and kriging variance at $\mathbf{u}$, respectively. Note that each mixture component in (3.9) may not be distinct; in the case where $j(\ell) = j(\ell')$, the ℓth and ℓ'th components are identical. When $j(1) = \cdots = j(L) = J$ (i.e., when $\mathbf{u} \in \text{Int}(\mathcal{S}_J)$), (3.9) reduces to the same predictive distribution given by (3.3), suggesting that $\mathbf{u}$ can be assigned to the Jth cluster with high certainty.

In general, the number of neighbors L controls the smoothness of the kriging mean at $\mathbf{u}$ near the boundary set $\mathcal{D} \setminus \cup_{j=1}^{k} \text{Int}(\mathcal{S}_j)$. As L increases, we have a larger boundary set, allowing for capturing partitioning uncertainty in a larger area, and the smoothing effects are stronger for locations within the boundary set. See Figure B.2 in Supplementary Section B.3.1 for an illustration of the kriging means and standard deviations (SDs) across $\mathcal{D} = [0, 1]^2$ with various values of L and a discrete uniform distribution for neighbor choices.

Example 3.2. Consider the $L = 1$ case. Then the SPGP becomes a piecewise GP in the sense that it takes the form $\text{GP}(0, C(\cdot, \cdot|\boldsymbol{\theta}_j))$ in the unioned region $\cup_{\mathbf{s}\in\mathcal{S}_j} V_{\mathcal{S}}(\mathbf{s})$, where $V_{\mathcal{S}}(\mathbf{s}) = \{\mathbf{v} \in \mathcal{D} : d(\mathbf{v}, \mathbf{s}) < d(\mathbf{v}, \mathbf{s}')$ for any $\mathbf{s}' \in \mathcal{S}$ and $\mathbf{s}' \neq \mathbf{s}\}$ is the Voronoi cell with nucleus $\mathbf{s}$ for the Voronoi tessellation based on $\mathcal{S}$. In particular, when each $\mathcal{S}_j$ is a singleton, the SPGP contains the piecewise GP based on Voronoi tessellations (Kim et al., 2005) as a special case. In our framework, a generic partition model is allowed for $\mathcal{S}_j$ so that the Voronoi cells with nuclei in $\mathcal{S}_j$ can be merged freely, leading to a piecewise GP with a more flexible space partition than the model in Kim et al. (2005).

3.3 Bayesian Spanning-Treed Gaussian Process Models

3.3.1 A Predictive Spanning-Treed Prior for Partitions

The SPGP defined in Section 3.2.3 is conditional on a partition $\pi_k(\mathcal{S})$. In a Bayesian modeling framework, $\pi_k(\mathcal{S})$ is treated as unknown and assigned a prior model. As we stressed in Introduction, it is desired to impose contiguity constraints for spatial partitioning problems. Henceforth, we will focus on spatially contiguous partitions and refer to them simply as partitions when there is no risk of ambiguity.

Instead of directly modeling $\pi_k(\mathcal{S})$, we consider a more general approach that builds partitions on a latent set to enable dimension reductions, referred to as the *predictive spanning-*

treed partition prior, and prove its richness in characterizing contiguous partitions. More precisely, we let $\mathcal{S}^* = \{\mathbf{s}_1^*, \ldots, \mathbf{s}_m^*\}$ be a set of pre-specified *reference knots*, which may or may not coincide with $\mathcal{S}$. We assume that $\pi_k(\mathcal{D})$ is a partition of the domain obtained from assigning $\mathbf{s} \in \mathcal{D}$ to the same cluster as its nearest neighbor in $\mathcal{S}^*$ under $\pi_k(\mathcal{S}^*)$. That is, we have $\mathcal{D}_j = \mathcal{D} \cap \left(\cup_{\mathbf{s}^* \in \mathcal{S}_j^*} V_{\mathcal{S}^*}(\mathbf{s}^*) \right)$. A partition $\pi_k(\mathcal{S})$ can be derived by setting $\mathcal{S}_j = \mathcal{S} \cap \mathcal{D}_j$. See Figure 3.1(b, c) for an illustration on how $\pi_k(\mathcal{D})$ and $\pi_k(\mathcal{S})$ are determined by $\pi_k(\mathcal{S}^*)$. As will be discussed in Section 3.4.3, this formulation provides a natural framework to achieve scalability for large spatial data sets by choosing $m \ll n$. The prior models on $\pi_k(\mathcal{D})$ and $\pi_k(\mathcal{S})$ can therefore be induced by a prior model on $\pi_k(\mathcal{S}^*)$.

Motivated by the success of spanning tree models for capturing contiguous partitions in linear regression settings (see, e.g., Luo et al., 2021b), we assign a spanning tree based partitioning prior for $\pi_k(\mathcal{S}^*)$. This prior has the advantages that (i) its support is rich enough to accommodate all possible spatially contiguous partitions of $\mathcal{S}^*$ (see also Proposition 3.3), (ii) it allows the number of clusters to be determined by the data, and (iii) it can facilitate computation by simplifying a complicated combinatorial partitioning problem on a graph into a compact representation based on spanning trees.

Let $\mathcal{G} = (\mathcal{S}^*, \mathcal{E})$ be an undirected graph with vertex set $\mathcal{S}^*$ and edge set $\mathcal{E} \subseteq \{(\mathbf{s}_i^*, \mathbf{s}_{i'}^*) : i \neq i'\}$, where $(\mathbf{s}_i^*, \mathbf{s}_{i'}^*)$ is an unordered set. Guided by our theoretical results (see Assumption SD in Section 3.3.3), $\mathcal{G}$ is specified as a radius-based nearest neighbor (R-NN) graph or a Delaunay triangulation with edges longer than a threshold removed such that locations connected by an edge are spatially adjacent. An example of a Delaunay triangulation graph is shown in Figure 3.1(a).

A spanning tree of $\mathcal{G}$ is defined as a subgraph $\mathcal{T} = (\mathcal{S}^*, \mathcal{E}_\mathcal{T}), \mathcal{E}_\mathcal{T} \subseteq \mathcal{E}$ that connects all vertices without any cycle. Let $\omega_{i,i'}$ be the weight of the edge $(\mathbf{s}_i^*, \mathbf{s}_{i'}^*)$ in $\mathcal{G}$ and $\boldsymbol{\omega} = \{\omega_{i,i'} : (\mathbf{s}_i^*, \mathbf{s}_{i'}^*) \in \mathcal{E}\}$. A minimum spanning tree (MST) is a spanning tree that has minimal sum of edge weights $\sum_{(\mathbf{s}_i^*, \mathbf{s}_{i'}^*) \in \mathcal{E}_\mathcal{T}} \omega_{i,i'}$.

A well-known property of spanning trees is that after a set of $k-1$ edges is removed from

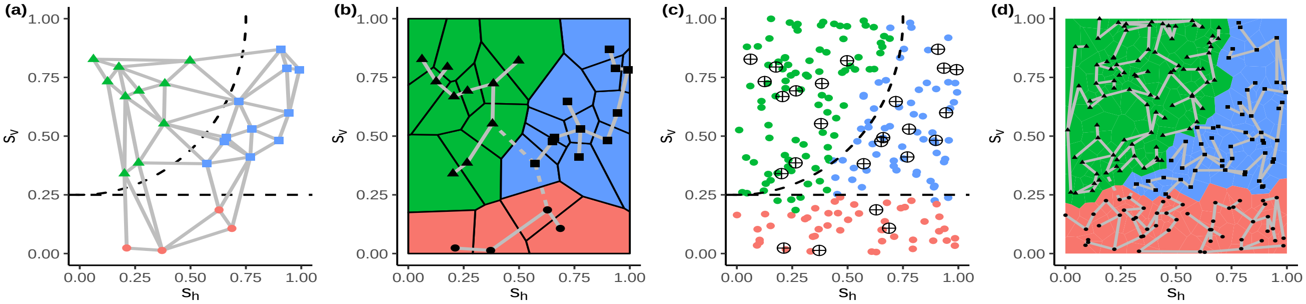

Figure 3.1: (a) A Delaunay triangulation graph on reference knots $\mathcal{S}^*$. The true space partition is represented by the dashed lines. (b) A spanning tree of the graph (grey lines) and a partition $\pi_3(\mathcal{S}^*)$ and the corresponding Voronoi cells on $\mathcal{S}^*$ induced by removing the two dashed edges. (c) A partition $\pi_3(\mathcal{S})$ of observed locations $\mathcal{S}$ (marked by dots) induced by $\pi_3(\mathcal{S}^*)$ (black crosses). (d) A space partition $\pi_3(\mathcal{D})$ when $\mathcal{S}^*$ is set to be $\mathcal{S}$.

$\mathcal{T}$, we obtain a graph with k connected subgraphs. By treating the jth connected subgraph as cluster $\mathcal{S}_j^*$, we obtain a spatially contiguous partition $\pi_k(\mathcal{S}^*)$. We say $\pi_k(\mathcal{S}^*)$ is *induced* by $\mathcal{T}$ in this case. See Figure 3.1(b) for an example of $\pi_3(\mathcal{S}^*)$ induced from a spanning tree. The estimation of $\pi_k(\mathcal{S}^*)$ amounts to learning the spanning tree (may not be unique) and its removed edges that induce the true partition. This property implies that a prior on $\pi_k(\mathcal{S}^*)$ can be assigned hierarchically, by first placing priors on the number of clusters k and the spanning trees in $\mathcal{G}$, and then the positions of the $k-1$ removed edges.

Formally, conditional on $\mathcal{T}$ and k we assume a uniform prior on all possible partitions induced by $\mathcal{T}$:

$$p\left\{\pi_k(\mathcal{S}^*) \mid k, \mathcal{T}\right\} \propto \mathbb{1}\left\{\pi_k(\mathcal{S}^*) \text{ is induced by } \mathcal{T} \text{ and has } k \text{ clusters}\right\}. \tag{3.10}$$

Regarding the prior on $\mathcal{T}$, a seemingly natural choice is to assume a discrete uniform distribution on all possible spanning trees of $\mathcal{G}$. However, it is challenging to sample from this uniform distribution. We opt to place an iid uniform prior on edge weights $\boldsymbol{\omega}$ instead, which induces a prior model on the spanning tree space via

$$\mathcal{T} = \text{MST}(\boldsymbol{\omega}), \quad \omega_{i,i'} \overset{\text{iid}}{\sim} \text{Unif}\,(0,1), \tag{3.11}$$

where $\mathrm{MST}(\boldsymbol{\omega})$ means an MST of the graph $\mathcal{G}$ based on edge weights $\boldsymbol{\omega}$. This MST space constructed from random edge weights consists of all possible spanning trees of $\mathcal{G}$. We will show in Section 3.4 that this prior also leads to an exact and fast spanning tree sampler, taking advantage of the Prim's algorithm for MST constructions.

Finally, we assume a truncated geometric distribution on k such that

$$\mathbb{P}(k = j) \propto (1 - c)^j, \quad \text{for } j = 1, \ldots, \bar{k}_m,\ 0 \leq c < 1, \tag{3.12}$$

where $\bar{k}_m$ is the pre-specified maximum number of clusters and c is a hyperparamter controlling the decaying rate of the prior probability so that models with large number of clusters can be penalized. Guided by our theoretical results in Section 3.3.3, we recommend specifying $\bar{k}_m$ such that it scales with $\sqrt{m \log m}$ (see Assumption P1). As discussed before, this prior on latent $\pi_k(\mathcal{S}^*)$ induces predictive spanning treed priors on $\pi_k(\mathcal{S})$ and $\pi_k(\mathcal{D})$. Below, we state two propositions concerning the supports of the prior models on a finite set $\mathcal{S}^*$ and an infinite set $\mathcal{D} = [0, 1]^d$, respectively. The definitions of contiguous partitions and proofs of both propositions are delayed to Appendix B.1.2. The first proposition states that the support of the predictive spanning-treed partition prior contains any spatially contiguous partitions on $\mathcal{S}^*$ (and hence $\mathcal{S}$ when $\mathcal{S} = \mathcal{S}^*$) with no more than $\bar{k}_m$ clusters.

Proposition 3.3. *Let $\pi_k(\mathcal{S}^*) = \{\mathcal{S}_1^*, \ldots, \mathcal{S}_k^*\}$ be an arbitrary spatially contiguous partition. Then $\pi_k(\mathcal{S}^*)$ is within the support of the prior defined by* (3.10)*,* (3.11)*, and* (3.12) *if $k \leq \bar{k}_m$.*

The next proposition shows that the predictive spanning-treed partition prior on the domain can approximate any fixed partition of $\mathcal{D}$ arbitrarily well as $\mathcal{S}^*$ becomes denser. As shown in Figure 3.1(d), if we choose a denser $\mathcal{S}^*$ (which is $\mathcal{S}$ in this example), the determined $\pi_k(\mathcal{D})$ better approximates the true partition (cf. Figure 3.1(b)).

Proposition 3.4. *Let $\pi_k(\mathcal{D}) = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ be an arbitrary fixed spatially contiguous partition of $\mathcal{D} = [0, 1]^d$. If $\mathcal{S}^*$ is a sequence of m locations distributed independently on $\mathcal{D}$ with a probability density function p_s such that $\inf_{\mathbf{s}^* \in \mathcal{D}} p_s(\mathbf{s}^*) > 0$ and $k \leq \bar{k}_m$, then there exists a* *partition* $\pi_k(\mathcal{S}^*) = \{\mathcal{S}_1^*, \ldots, \mathcal{S}_k^*\}$ *with positive prior probability such that*

$$\mathcal{L}\left\{\mathcal{D}_j \Delta \left(\cup_{\mathbf{s}^* \in \mathcal{S}_j^*} V_{\mathcal{S}^*}(\mathbf{s}^*)\right)\right\} \longrightarrow 0 \quad \text{for } j = 1, \ldots, k,$$

as $m \to \infty$ *almost surely under the data generating process of* $\mathcal{S}^*$*, where* $\mathcal{L}(\cdot)$ *denotes Lebesgue measure and* Δ *denotes symmetric difference of sets.*

3.3.2 Spanning-Treed Gaussian Process Regressions

We embed the proposed L-SPGP with a spanning-treed partition prior into a spatial regression setting. Consider a point-referenced response variable $y(\mathbf{s}) \in \mathbb{R}$ at a generic location $\mathbf{s} \in \mathcal{D}$ along with a vector of covariates $\mathbf{x}(\mathbf{s}) \in \mathbb{R}^p$. We denote the collection of responses and the design matrix corresponding to a generic finite subset $\mathcal{A}$ of $\mathcal{D}$ by $\mathbf{y}(\mathcal{A})$ and $\mathbf{X}(\mathcal{A})$, respectively.

We consider a spatial regression model specified as

$$y(\mathbf{s}) = \boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}(\mathbf{s}) + w(\mathbf{s}), \quad \mathbf{s} \in \mathcal{D},$$

where the residual process $w(\mathbf{s})$ is modeled as a zero-mean L-SPGP conditional on a partition $\pi_k(\mathcal{S})$, which is fully determined by $\pi_k(\mathcal{S}^*)$ as discussed in Section 3.3.1. Finally, we complete the hierarchical model by assigning a spanning-treed prior to $\pi_k(\mathcal{S}^*)$. We call this model the spanning-treed Gaussian process (STGP) regression. The hierarchical model of STGP for observations can be written as

$$\mathbf{y}(\mathcal{S}_j)|\boldsymbol{\beta}, \boldsymbol{\Theta}, \pi_k(\mathcal{S}) \overset{\text{ind.}}{\sim} \mathrm{N}_{n_j}\left\{\mathbf{X}(\mathcal{S}_j)\boldsymbol{\beta},\ \mathbf{C}(\mathcal{S}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)\right\}, \tag{3.13a}$$

$$\boldsymbol{\beta}|\lambda \sim \mathrm{N}_p\left(\boldsymbol{\mu}_\beta, \lambda\mathbf{I}_p\right), \quad \lambda \sim \mathrm{IG}(a_\lambda, b_\lambda), \tag{3.13b}$$

$$\big(\pi_k(\mathcal{S}^*), k, \mathcal{T}\big) \sim p\big(\pi_k(\mathcal{S}^*)|k, \mathcal{T}\big)p(k)p(\mathcal{T}), \tag{3.13c}$$

where $p(\pi_k|k, \mathcal{T})$, $p(\mathcal{T})$, and $p(k)$ are specified in (3.10), (3.11), and (3.12) respectively. Note that we assume all clusters share the same coefficients. One may instead assume cluster-specified coefficients; however, we argue that this may cause identifiability issues between spatially varying regression means and spatial random effects and hence a poor parameter estimation, though we can still obtain reasonable prediction accuracy of the responses.

We complete the hierarchical model by specifying the local covariance function. One popular choice is the *stationary* Matérn family (Banerjee et al., 2004) including both isotropic models $\sigma^2\rho(\mathbf{s},\mathbf{s}'|\phi,\nu)+\tau^2\mathbb{1}(\mathbf{s}=\mathbf{s}')$, where σ^2, ϕ, ν and τ^2 are the variance, range, smoothness and nugget effect variance parameters respectively, and geometric *anisotropic* models. Priors for local covariance parameters are assigned following standard GP models.

Finally, consider a new location $\mathbf{u} \notin \mathcal{S}$ where we intend to predict the response $y(\mathbf{u})$ given $\mathbf{x}(\mathbf{u})$ and $\mathbf{y}(\mathcal{S})$. Following (3.9), the posterior predictive distribution of $y(\mathbf{u})$ is

$$y(\mathbf{u})|\mathbf{y}(\mathcal{S}),\boldsymbol{\beta},\boldsymbol{\Theta},\pi_k(\mathcal{S}) \sim \sum_{\ell=1}^{L} \alpha_\ell \; \mathrm{N}_1\left(\tilde{\boldsymbol{\mu}}(\mathbf{u}|\mathcal{S}_{j(\ell)},\boldsymbol{\theta}_{j(\ell)}),\boldsymbol{\Sigma}(\mathbf{u}|\mathcal{S}_{j(\ell)},\boldsymbol{\theta}_{j(\ell)})\right), \tag{3.14}$$

with $\tilde{\boldsymbol{\mu}}(\mathcal{U}|\mathcal{S},\boldsymbol{\theta}) = \mathbf{X}(\mathcal{U})\boldsymbol{\beta} + \mathbf{C}(\mathcal{U},\mathcal{S}|\boldsymbol{\theta})\mathbf{C}^{-1}(\mathcal{S},\mathcal{S}|\boldsymbol{\theta})\left\{\mathbf{y}(\mathcal{S}) - \mathbf{X}(\mathcal{S})\boldsymbol{\beta}\right\}.$

3.3.3 Theoretical Properties

In this subsection we establish posterior concentration results for the STGP regression model under the assumption that $\mathcal{D} = [0,1]^2$ and the true spatial field is a piecewise smooth function. Our theoretical results can be easily extended to a more general domain that is homeomorphic to the unit square with the Euclidean metric and a bi-Lipschitz homeomorphism. Throughout this subsection, we focus on the case where $\mathcal{S}^* = \mathcal{S}$. Assuming $y(\mathbf{s})$ has zero mean for simplicity, our model can be written as

$$y(\mathbf{s}) = \tilde{w}(\mathbf{s}) + \varepsilon(\mathbf{s}), \quad \varepsilon(\mathbf{s}) \sim \mathrm{N}_1\left\{0, \tau^2(\mathbf{s})\right\} \tag{3.15}$$

where $\tilde{w}(\mathbf{s})$ is assigned a spanning-treed isotropic GP prior with local Matérn parameters $\{\sigma_j^2, \phi_j\}$ and a common smoothness parameter ν, and $\varepsilon(\mathbf{s})$ is the nugget effect with a piecewise constant variance $\{\tau_j^2\}$.

We adopt the following notations. Given two positive sequences $\{a_n\}$ and $\{b_n\}$, $a_n = o(b_n)$ means $\lim_{n\to\infty}(a_n/b_n) = 0$ and $a_n \asymp b_n$ means $0 < \liminf_{n\to\infty}(a_n/b_n) \le \limsup_{n\to\infty}(a_n/b_n) < \infty$. The posterior given data $(\mathbf{y}(\mathcal{S}), \mathcal{S})$ is denoted by $\Pi_n(\cdot|\mathbf{y}(\mathcal{S}), \mathcal{S})$.

We first state the assumptions on the true data generating process. We assume the responses are generated according to (3.15) with a piecewise smooth true mean function $\tilde{w}^*(\mathbf{s})$ and a piecewise constant true nugget variance $\tau^{*2}(\mathbf{s})$. More precisely, we let $\pi^*_{k^*}(\mathcal{D}) = \{\mathcal{D}^*_1, \ldots, \mathcal{D}^*_{k^*}\}$ be the true contiguous partition of $[0,1]^2$ with some fixed k^* and a fixed boundary set $\mathcal{B}^* \subset [0,1]^2$ (see Supplementary Section B.1.2 for the definition). We assume the following smoothness conditions on the true spatial field in each $\mathcal{D}^*_j$.

Assumption T. *We assume $\tilde{w}^*(\mathbf{s})$ and $\tau^{*2}(\mathbf{s})$ satisfy*

$$\tilde{w}^*(\mathbf{s}) = \sum_{j=1}^{k^*} \tilde{w}^*_j(\mathbf{s}) \mathbb{1}(\mathbf{s} \in \mathcal{D}^*_j), \quad \tau^*(\mathbf{s}) = \sum_{j=1}^{k^*} \tau^*_j \mathbb{1}(\mathbf{s} \in \mathcal{D}^*_j),$$

for some functions $\tilde{w}^*_j \in C^\beta[0,1]^2 \cap H^\beta[0,1]^2$ and constants $\tau^*_j > 0$ that are fixed as n grows, where $C^\beta[0,1]^2$ and $H^\beta[0,1]^2$ are the Hölder space and the Sobolev space of regularity β, respectively. Further, we assume that $\tilde{w}^*_j(\cdot)$ is within the support of a GP prior with an isotropic Matérn covariance $\sigma^{2*}_j \rho(\cdot,\cdot|\phi^*_j, \nu)$ for some constants σ^{*2}_j, ϕ^*_j, and a known $\nu \ge \beta$.

We adopt a random design framework where the number of sampling locations within a fixed domain diverges to infinity. We assume the following on the spatial design and spatial graph of n points $\mathbf{s}_1, \ldots, \mathbf{s}_n$ in $\mathcal{D}$.

Assumption SD. *Given $n \in \mathbb{N}$, we assume $\mathcal{S}$ is a sequence of n independent points where each point is distributed on $[0,1]^2$ with a probability density function p_s such that $0 < p^{\min}_s \le p_s(\mathbf{s}) \le p^{\max}_s < \infty$. We assume the spatial graph on $\mathcal{S}$ is constructed by (i) the R-NN graph with a radius $\gamma_1 \asymp \sqrt{\log n/n}$ and $\gamma_1 > \gamma_0$, where γ_0 is the maximum edge length of the MST on $\mathcal{S}$; or (ii) the Delaunay triangulation graph where the edges are removed if they are longer than γ_2, where $\gamma_2 \asymp \sqrt{\log n/n}$ and $\gamma_2 > \gamma_0$.*

Given the true space partition and a spatial graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, we say an edge $(\mathbf{s}_i, \mathbf{s}_{i'}) \in \mathcal{E}$ is across the true boundary $\mathcal{B}^*$ if $\mathbf{s}_i \in \mathcal{D}_j^*$ and $\mathbf{s}_{i'} \in \mathcal{D}_{j'}^*$ for some $j \neq j'$. If the set of all edges within a spanning tree $\mathcal{T}$ that are across $\mathcal{B}^*$ is removed, one obtains a partition of $\mathcal{S}$, denoted by $\pi^*_{k^*_{\mathcal{T}}}(\mathcal{S})$, that is nested in the true partition $\pi^*_{k^*}(\mathcal{S}) = (\mathcal{S} \cap \mathcal{D}_1^*, \ldots, \mathcal{S} \cap \mathcal{D}_{k^*}^*)$ of $\mathcal{S}$, and with the number of clusters $k^*_{\mathcal{T}} \geq k^*$. Assumption SD guarantees that the maximum number of edges across $\mathcal{B}^*$ in any spanning tree scales with $\sqrt{n \log n}$ with probability tending to 1 (Luo et al., 2021b). This implies $k^*_{\mathcal{T}} \leq c_1\sqrt{n \log n}$ for some constant $c_1 > 0$ and any $\mathcal{T}$. This bound plays a crucial role in establishing the prior concentration around the true model.

We further assume the priors satisfy the following condition, which guarantees the partition $\pi^*_{k^*_{\mathcal{T}}}(\mathcal{S})$ is within the support of the prior given an arbitrary spanning tree $\mathcal{T}$. It also regularizes the partition model so that the number of obtained clusters is not too large.

Assumption P1. *We assume $\bar{k}_n$ satisfies $c_1\sqrt{n \log n} \leq \bar{k}_n \leq c_1'\sqrt{n \log n}$ for some constants $c_1' > c_1$.*

We are now ready to state our first posterior concentration result. We denote by $p(y|\mathbf{s})$ the conditional density of the response given the sampled location, whereas the true one is denoted by $p^*(y|\mathbf{s})$. Note that $p(y|\mathbf{s})$ depends on the partition and covariance parameters. The following theorem shows that $p(y|\mathbf{s})$ concentrates in a weak neighborhood of $p^*(y|\mathbf{s})$ asymptotically under a random spatial design for $\mathcal{S}$. Its proof is deferred to Appendix B.1.4.

Theorem 3.5 (Weak consistency)**.** *Define the weak ϵ-neighborhood of true density $p^*(y|\mathbf{s})$ for any bounded continuous function g and any $\epsilon > 0$ as*

$$W_{g,\epsilon} = \left\{ p : \left| \int g(y|\mathbf{s})p(y|\mathbf{s})p_s(\mathbf{s})dyd\mathbf{s} - \int g(y|\mathbf{s})p^*(y|\mathbf{s})p_s(\mathbf{s})dyd\mathbf{s} \right| < \epsilon \right\}.$$

Under Assumptions T, SD, and P1, the posterior distribution satisfies $\Pi_n\left(W^c_{g,\epsilon} \mid \mathbf{y}(\mathcal{S}), \mathcal{S}\right) \to 0$ almost surely under $p^*(y|\mathbf{s})p_s(\mathbf{s})$.

To establish posterior contraction rate results, we need additional assumptions on the priors and the spatial graph. Let ϵ_n be a sequence going to zero such that $\epsilon_n \asymp (\log n/n)^\delta$ with some constant $0 < \delta < \min\{\beta/(8\nu + 8 - 4\beta), 1/4 - 1/(2\alpha)\}$, where $\alpha = \lfloor \nu \rfloor$.

Assumption P2.

(P2-1) Assume that $\nu \geq \max(3, \beta)$.

(P2-2) There exist sequences $\tilde{\phi}_n$, $\tilde{\sigma}_n$ *and* M_n *satisfying that, as* $n \to \infty$,

$$-\log \Pi_\phi(\phi < \tilde{\phi}_n^{-1})/(n\epsilon_n^2) \to +\infty, \quad -\log \Pi_\sigma(\sigma^2 > \tilde{\sigma}_n^{-2})/(n\epsilon_n^2) \to +\infty,$$

$$\bar{k}_n(M_n/\epsilon_n)^{2/\alpha} = o(n\epsilon_n^2), \quad M_n^2 \tilde{\sigma}_n^2 \tilde{\phi}_n^{-2\alpha}/(n\epsilon_n^2) \to +\infty.$$

(P2-3) Π_τ *is supported on* $[a, b] \subset \mathbb{R}$ *with* $0 < a \leq \tau_j^{*2} \leq b < +\infty$ *for all* $j = 1, \ldots, k^*$.

Assumption SG. *Let* $\xi_n(k)$ *be the number of unique spatially contiguous partitions with* k *clusters of the graph* $\mathcal{G}$ *on* $\mathcal{S}$*. We assume* $\mathcal{G}$ *is constructed such that* $\log\left(\max_{1\leq k\leq \bar{k}_n} \xi_n(k)\right) = O(n\epsilon_n^2)$.

Assumptions (P2-1) and (P2-2) on the priors of covariance functions allow us to construct a sieve on $\tilde{w}$ that has desired tail probability and metric entropy; similar assumptions can be found in Ghosal and Roy (2006) and Payne et al. (2020). Assumption (P2-3) is a standard assumption in the literature for nonparametric regressions with GP priors (see van der Vaart and van Zanten, 2008; Bhattacharya et al., 2014, among others), which is used to construct a sieve on τ^2. Assumption SG excludes some graphs that are too dense and constrains the complexity of the space of all possible partitions so that the test functions with desired probability of type-I errors exist.

The next theorem suggests that the posterior contracts with rate ϵ_n at $p^*(y|\mathbf{s})$ with respect to expected total variation distance. Note that this rate is slower than the minimax rate for customary GP regressions with Matérn kernels (van der Vaart and van Zanten, 2011) as we pay a price for estimating the unknown partition structure using the flexible spanning-treed prior. The detailed proof is provided in Appendix B.1.5.

Theorem 3.6 (Posterior contraction)**.** *Under the same assumptions in Theorem 3.5 as well as Assumptions P2 and SG, the posterior distribution satisfies*

$$\Pi_n\left(\int |p(y|\mathbf{s}) - p^*(y|\mathbf{s})|\, p_s(\mathbf{s}) dy d\mathbf{s} \geq M\epsilon_n \mid \mathbf{y}(\mathcal{S}), \mathcal{S}\right) \longrightarrow 0$$

almost surely under $p^*(y|\mathbf{s})p_s(\mathbf{s})$ for some constant $M > 0$.

3.4 Computational Strategies

3.4.1 Estimation

The unknown parameters of the proposed STGP regression model mainly involve the spanning-treed partition parameters $\big(\pi_k(\mathcal{S}), k, \mathcal{T}\big)$, the associated cluster-specified covariance parameters $\mathbf{\Theta} = \big\{\tau_j^2, \sigma_j^2, \tilde{\boldsymbol{\theta}}_j\big\}_{j=1:k}$ with local correlation parameters $\tilde{\boldsymbol{\theta}}_j$, and the global parameters $(\boldsymbol{\beta}, \lambda)$. Conditional on $\big(\pi_k(\mathcal{S}), k, \mathcal{T}\big)$ and $\mathbf{\Theta}$, global parameters can be updated via standard Bayesian inference methods. In particular, we sample $\boldsymbol{\beta}$ and λ from their posterior conditional distributions, which follow a multivariate normal and an inverse gamma distribution, respectively. The detailed forms are included in Supplementary Section B.2.

Below, we focus on the adaptive estimation of spanning treed partitions $\big(\pi_k(\mathcal{S}), k, \mathcal{T}\big)$ and covariance parameters $\mathbf{\Theta}$ conditional on $(\boldsymbol{\beta}, \lambda)$. As the number of clusters is assumed unknown, this trans-dimensional inference is done via a tailored reversible jump Markov chain Monte Carlo (RJ-MCMC) sampler (Green, 1995; Luo et al., 2021b). Taking advantage of the tree structure, each RJ-MCMC move can be achieved by simply adding and/or deleting an edge in the tree, or updating trees via efficient MST algorithms. The acceptance ratio of the proposed RJ-MCMC move involves the calculation of likelihood ratios, a major computation bottleneck in standard RJ-MCMC algorithms. We will show that, each RJ-MCMC move under STGP only changes the cluster memberships of a smaller subset of observations, and hence only the likelihood ratios involving this subset of data need to be calculated. An additional advantage of the locally stationary model is that it allows to estimate cluster-specific parameters $\mathbf{\Theta}$ using only the data in each subregion. In doing so, STGP naturally leads to a reduced computation from fitting a global GP model to a number of local GP models.

Specifically, we reparametrize the covariance function by setting $\sigma_j^2 = \tau_j^2 \bar{\sigma}_j^2$ and place a conjugate inverse Gamma prior for $\boldsymbol{\tau}^2 = \{\tau_j^2\}_{j=1:k}$ that allows us to integrate τ_j^2 out analytically when we update the partitions and other cluster-specific parameters, which improves mixing and convergence of our sampler.

To collect samples from $\big(\{\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j\}_{j=1:k}, \pi_k(\mathcal{S}), k, \mathcal{T}\big)|(\boldsymbol{\tau}^2, \boldsymbol{\beta}, \lambda)$, one of the four moves — *birth*, *death*, *change*, and *hyper* — is performed with probabilities $r_b(k)$, $r_d(k)$, $r_c(k)$, and $r_h(k)$, respectively. The first three moves modify the partition $\pi_k(\mathcal{S}^*)$, which in turn determines a modification of $\pi_k(\mathcal{S})$.

In a *birth* move, one of the k clusters in $\pi_k(\mathcal{S}^*)$ is randomly chosen with equal probabilities, and then the chosen cluster is split into two by randomly removing an edge in $\mathcal{T}$ that connects vertices in the cluster. Suppose that $\mathcal{S}_{j_0}^*$ is chosen to be split into $\mathcal{S}_{j_1}^*$ and $\mathcal{S}_{j_2}^*$. In the case where $\mathcal{S}^* \neq \mathcal{S}$, $\mathcal{S}_{j_0}$ is also split into two clusters $\mathcal{S}_{j_1}$ and $\mathcal{S}_{j_2}$, by assigning $\mathbf{s} \in \mathcal{S}_{j_0}$ to $\mathcal{S}_{j_1}$ (or $\mathcal{S}_{j_2}$) if its nearest neighbor in $\mathcal{S}^*$ belongs to $\mathcal{S}_{j_1}^*$ (or $\mathcal{S}_{j_2}^*$). One of $\mathcal{S}_{j_1}$ and $\mathcal{S}_{j_2}$ is uniformly chosen to inherit the parameters $(\bar{\sigma}^2, \tilde{\boldsymbol{\theta}})$ from the original cluster. As there is no conjugate prior for $\bar{\sigma}^2$ or $\tilde{\boldsymbol{\theta}}$, standard Metropolis-Hastings (M-H) updates can lead to low efficiency. To address this, following Payne et al. (2020), the $(\bar{\sigma}^2, \tilde{\boldsymbol{\theta}})$ for the other new cluster, say $\mathcal{S}_{j_2}$, are chosen to maximize $p\left\{\mathbf{y}(\mathcal{S}_{j_2})|\bar{\sigma}^2, \tilde{\boldsymbol{\theta}}, -\right\} p(\bar{\sigma}^2)p(\tilde{\boldsymbol{\theta}})$, where $p(\bar{\sigma}^2)$ and $p(\tilde{\boldsymbol{\theta}})$ are the prior densities for $\bar{\sigma}^2$ and $\tilde{\boldsymbol{\theta}}$, respectively, and $p\left\{\mathbf{y}(\mathcal{S}_j)|\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j, -\right\}$ is the likelihood function of $\mathbf{y}(\mathcal{S}_j)$ with τ^2 integrated out. The M-H ratio is therefore

$$(1-c) \times \frac{r_d(k+1)}{r_b(k)} \times \frac{p\left\{\mathbf{y}(\mathcal{S}_{j_1})|\bar{\sigma}_{j_1}^2, \tilde{\boldsymbol{\theta}}_{j_1}, -\right\} p\left\{\mathbf{y}(\mathcal{S}_{j_2})|\bar{\sigma}_{j_2}^2, \tilde{\boldsymbol{\theta}}_{j_2}, -\right\}}{p\left\{\mathbf{y}(\mathcal{S}_{j_0})|\bar{\sigma}_{j_0}^2, \tilde{\boldsymbol{\theta}}_{j_0}, -\right\}}, \tag{3.16}$$

which only involves likelihood functions on subsets of $\mathcal{S}$.

Opposite to the *birth* move, a *death* move randomly merges two adjacent clusters in $\pi_k(\mathcal{S}^*)$. Specifically, an edge in $\mathcal{T}$ that connects two distinct clusters in $\pi_k(\mathcal{S}^*)$ is uniformly selected and then the two clusters are merged. The corresponding two clusters in $\pi_k(\mathcal{S})$ are also merged accordingly. The parameters $(\bar{\sigma}^2, \tilde{\boldsymbol{\theta}})$ of the merged clusters are chosen using a similar maximum a posteriori (MAP) approach as in the birth move. The M-H ratio is analogous to (3.16). In a *change* move, a *death* move is performed followed by a *birth* move, so that the number of clusters is unchanged. This move is designed to encourage better mixing of the Markov chain.

A *hyper* move updates the spanning tree using the exact sampler similar as in Luo et al. (2021b), which adaptively learns a desired spanning tree spatial order to better recover the true partition. We sample the edge weight $\omega_{i,i'}$ of $\mathcal{G}$ from iid Unif$(1/2, 1)$ if the vertices $\mathbf{s}_i^*$ and $\mathbf{s}_{i'}^*$ are in different clusters under $\pi_k(\mathcal{S}^*)$, and otherwise from iid Unif$(0, 1/2)$. A new spanning tree is the MST generated by Prim's algorithm using the new edge weights.

Finally we update the parameters $\{\tau_j^2\}_{j=1:k}$ by sampling from their inverse gamma full conditionals, whose closed forms are given in Supplementary Section B.2.

3.4.2 Prediction

Posterior predictive inference in the STGP model can be achieved via (3.14). Let $\mathcal{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ be a collection of locations where the responses are unobserved, i.e., $\mathbf{u}_i \notin \mathcal{S}$ for $i = 1, \dots, r$. Conditional on a posterior draw of the parameters, we can sample from $\mathbf{y}(\mathcal{U}) | \big(\mathbf{y}(\mathcal{S}), \boldsymbol{\Theta}, \pi_k(\mathcal{S}), k, \mathcal{T}\big)$. The detailed algorithm is provided in Supplementary Section B.2. Note that the prediction algorithm is parallelizable, as predictions at each $\mathcal{U}_j$ are independent and only depend on the observations from one subregion at a time given a sample of cluster memberships.

Thanks to the Gaussian mixture structure in the predictive distributions, the prediction uncertainty at points near boundaries will be reflected by their oftentimes multi-modal predictive distributions. As discussed in Section 3.2.3, the kriging mean predictive surface around the estimated boundary becomes smoother as L grows. The surface can be further smoothed by using Bayesian model averaging to account for model estimation uncertainties (Gramacy and Lee, 2008). We remark that the usual kriging means and SDs estimates may not be the ideal choice to summarize the possibly multi-modal spatial prediction results of the STGP model at boundaries. Instead, we recommend to use the highest posterior density (HPD) region to capture multimodality by disjoint HPD intervals.

3.4.3 Computation for Large Data Sets

Despite of the several advantages of the aforementioned tailored RJ-MCMC algorithm, there are still two major computational bottlenecks that need to be mitigated for very large spatial data sets.

First, the MCMC algorithm described in Section 3.4.1 involves graph operations that can be time-consuming when we set $\mathcal{S}^* = \mathcal{S}$ due to the large graph. A larger graph is also associated with a larger spanning treed partition space that may cause slower convergence and mixing of MCMC. We can mitigate this by specifying $\mathcal{S}^*$ of size $m \ll n$, which allows us to perform graph operations on a graph with fewer vertices and edges. The time complexity is $O(m)$ for a birth step in the MCMC algorithm and $O(m \log m)$ for a hyper step. Possible choices of $\mathcal{S}^*$ include regular grids and random subsets of the observed locations from k-means clustering or k-d tree partitions.

Similar to customary GP models, another major computational challenge of the STGP models comes from the cubic time complexity of matrix operations. Despite that the MCMC procedure only involves solving the linear system with a local covariance matrix $\mathbf{C}(\mathcal{S}_j, \mathcal{S}_j | \boldsymbol{\theta}_j)$, computation may still be an issue when a cluster contains a large amount of observations. Fortunately, the induced prior model on $\pi_k(\mathcal{S})$ provides a natural framework to incorporate block-based likelihood approximation methods. For each knot $\mathbf{s} \in \mathcal{S}^*$, the Voronoi cell $V_{\mathcal{S}^*}(\mathbf{s}^*)$ not only merges to form a partition on $\mathcal{S}$, but defines a block in the domain (see also Figure 3.1(b)). Using this blocking scheme, recent block-based scalable GP methods, such as the block version of NNGP methods (Datta et al., 2016; Zhang et al., 2019), full-scale approximations with blocks (Konomi et al., 2014) and the meshed GP method (Peruzzi et al., 2020), can be conveniently embedded into the current algorithm to speed up the local likelihood calculation by grouping observations within a Voronoi cell into a block. The stationary assumption underpinning some of these computation approximation algorithms now holds locally, so we expect that they would achieve a good approximation.

Finally, we remark that the reduced graph and blocking scheme can also provide a warm initialization of the original graph and the spanning tree at $\mathcal{S}^*$ to facilitate MCMC convergence. Specifically, one can fit an independent GP regression model within each Voronoi cell to obtain initial posterior parameter estimates, and then construct a data-driven graph by using the distance between the two initial posterior distributions of local parameters from the reference knots' corresponding Voronoi cells. This step can be naturally handled in a parallel fashion.

3.5 Simulation Studies

In this section, we assess the performance of the STGP regression model by some simulated data. We consider a squared spatial domain $\mathcal{D} = [0,1]^2$ that is partitioned into two regions $\mathcal{D}_1^*$ and $\mathcal{D}_2^*$ with the boundary given by a circle of radius 0.3 centered at $(0.5, 0.5)$. We generate $n = 500$ spatial locations $\mathcal{S}$ uniformly in $[0,1]^2$ for training data. To examine prediction performance, we also generate $r = 100$ hold-out locations $\mathcal{U}$ in the following way: with probability 0.75 a hold-out location is generated uniformly in a ring $\{(s_h, s_v) \in [0,1]^2 : 0.2^2 < (s_h - 0.5)^2 + (s_v - 0.5)^2 < 0.4^2\}$; with probability 0.25 we draw locations uniformly in its complement. This sampling scheme allows us to assess the prediction performance primarily at the locations near the true boundary where the abrupt changes happen. See Figure 3.2(a) for the sampled locations.

The responses are generated from (3.15) using isotropic Matérn covariance functions, where the true parameters of the processes in $\mathcal{D}_1^*$ and $\mathcal{D}_2^*$ have well-separated microergodic parameters $\vartheta = \sigma^2/\phi^{2\nu}$, and ν is treated as known. It is shown in Zhang (2004) that ϑ matters more in prediction and can be consistently estimated, while σ^2 and ϕ cannot. We specify the reference knots $\mathcal{S}^* = \mathcal{S}$. We follow the theoretical results in Section 3.3.3 to choose spatial graphs and priors for partitions. The detailed true parameter values, prior specifications and other model choices can be found in Supplementary Section B.3.2.

In both studies we compare the STGP model with treed Gaussian process (TGP) models (Gramacy and Lee, 2008), nonstationary Gaussian process (NSGP) models developed in Paciorek and Schervish (2006) and Risser and Turek (2020), and stationary Gaussian process (SGP) spatial regressions with isotropic Matérn covariance functions (see, e.g., Banerjee et al., 2014).

We run MCMC algorithms for each model for $30,000$ iterations, discarding the first half, and thin the chains every 10 iterations, yielding $1,500$ posterior draws for inference.

We first examine partition recovery performance between STGP and TGP. Figure 3.2 shows the MAP estimates of the partition. Due to the use of binary trees, TGP gives 4 rectangular clusters that do not match the true ones. The partition given by STGP, on the other hand, is fairly consistent to the truth considering that the estimation results are based on just one realization of the random field. For instance, the true cluster inside the true boundary is successfully recovered by Clusters 3 and 4 in Figure 3.2(a). This is also evidenced by the higher in-sample adjusted Rand indices (ARIs; Hubert and Arabie, 1985) in the first row of Table 3.1. Note that the in-sample ARI for STGP does not depend on the choice of L. The ARIs for the hold-out locations based on the MAP partition estimates are shown in the second row of Table 3.1. The STGP models with $L = 1, 3, 5$ have higher ARIs than TGP does, suggesting that the membership prediction from STGP agrees more with the true partition. For the STGP models, we also note that the ARI for the hold-out data is higher when $L = 1$, which is not surprising since locations close to each other tend to share the same cluster membership. However, as we will see later, setting $L = 3$ leads to better predictive performance although it may not has the best partition estimate.

Next, we consider estimation accuracy of covariance parameters measured by the mean square error of the MAP estimate of the log microergodic parameter $\log(\vartheta)$, denoted as MSE_ϑ. Note that MSE_ϑ for STGP does not depend on L. For SGP, $\hat{\vartheta}(\mathbf{s})$ reduces to a constant $\hat{\sigma}^2/\hat{\phi}^{2\nu}$. The resulting MSE_ϑ's are given in the third row of Table 3.1. The STGP model has the lowest estimation error, which means it can estimate the spatially varying

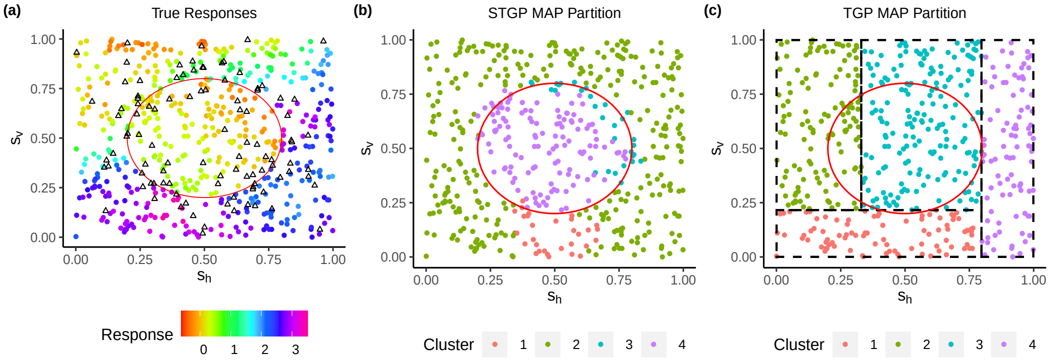

Figure 3.2: (a) True $y(s)$ of the training data sets. (b, c) MAP partition estimates given by STGP and TGP. The true boundary is marked by the red circle.

Table 3.1: Performance metrics of STGP and its competing methods.

	STGP ($L = 1$)	STGP ($L = 3$)	STGP ($L = 5$)	TGP	NSGP	SGP
In-sample ARI	**0.694**	**0.694**	**0.694**	0.367	—	—
Hold-out ARI	**0.485**	0.356	0.298	0.007	—	—
MSE$_\vartheta$	**6.555**	**6.555**	**6.555**	24.227	—	43.427
MSPE$_y$	0.198	0.130	0.139	**0.116**	0.236	0.138
Mean CRPS$_y$	0.159	**0.130**	0.135	0.156	0.207	0.186
Mean LogS$_y$	2.012	**-0.429**	-0.411	0.126	0.534	0.520

covariance parameters more accurately.

Finally, we analyze the performance of out-of-sample prediction. As shown in the fourth row of Table 3.1, the TGP model has the lowest mean squared prediction error (MSPE), followed by the STGP model with $L = 3$. We argue that, nonetheless, MSPE is not the most ideal metric to evaluate the performance of probabilistic prediction as it may not fully take into account the posterior predictive distributions. As a result, it is more sensible to compare scoring rules such as average CRPS and LogS (Gneiting and Raftery, 2007), which are presented in the last two rows in Table 3.1. The STGP models with $L = 3$ and 5 have the best scores overall among all models, while the one with $L = 1$ has a comparable CRPS as TGP does. The superior performance of the STGP models with $L = 3$ and 5 over the one with $L = 1$ is partly because they produce a smoother interpolation of the spatial field

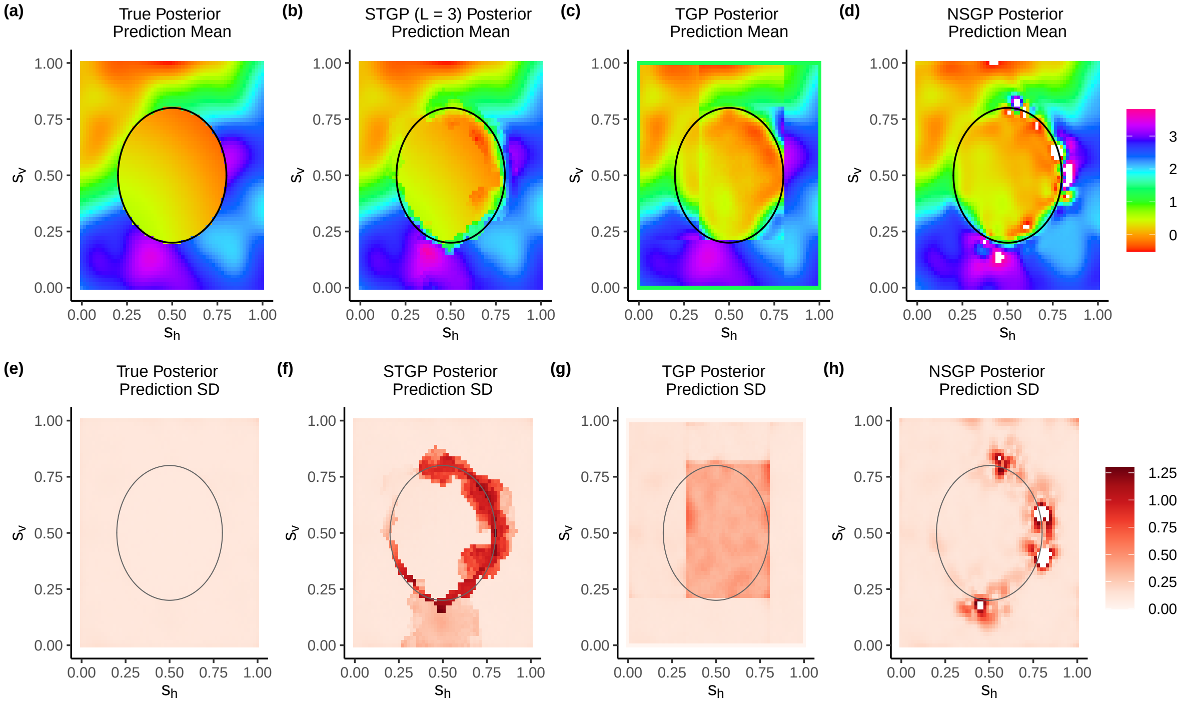

Figure 3.3: Posterior mean predictive surfaces (a-d) and SD surfaces (e-h) form the true data generating model, STGP with $L = 3$, TGP, and NSGP.

that is more robust to misclassification of cluster memberships. By setting $L > 1$ the model is more likely to correctly classify $\mathbf{u}$ with some positive probability when $\mathbf{u}$ does not belong to the cluster containing its nearest neighbor, allowing for better prediction performance and uncertainty quantification near the true boundary. We also observe that compared with CRPS, LogS is more sensitive to misclassification in the sense that a misclassified hold-out location near the true boundary can lead to a large LogS when $L = 1$.

Figure 3.3 displays posterior mean predictive surfaces and posterior prediction SDs from STGP with $L = 3$, TGP, and NSGP. We also include the kriging results from the model where the true partition and other parameters are known as a benchmark. The mean predictive surface from STGP closely approximates the true one. Due to the Gaussian mixture predictive distributions and Bayesian model averaging, we obtain a smooth surface near the true boundary rather than a sharp jump. As desired, the prediction SDs are higher around the true boundary, capturing the uncertainty from the unknown partition. Note that the locations near the top-left part of the true boundary have lower SDs, because of the relatively smaller jump in the true field in this region that reults in smaller uncertainty in prediction. In the surface from TGP, some discontinuities can be observed near the estimated boundaries, and some of them appear in the interior of a true cluster where the true surface is smooth. The SD plot from TGP provides little information for inferring the true boundary. Despite the NSGP predictive surface captures some patterns of the true field, it generates some poor predictions with fairly large or small values near the true boundary, and the high uncertainty region from it does not cover the bottom-left part of the true boundary. The results are not surprising because NSGP is better suited for the case where the change of covariance is relatively smooth. In contrast, the advantage of our method is more prominent when the true covariance function has abrupt changes or clustering patterns.

Since it is insufficient to visualize a possibly multi-modal posterior predictive distribution via its mean and SD, we further examine the plots of predictive densities for selected locations (see, e.g., Figure B.3 in Supplementary Section B.3.2). Our results confirm that STGP can quantify prediction uncertainty in a desirable way, where the higher mode appears near the true value and the corresponding 95% HPD interval also covers the true value. In contrast, the posterior predictive densities from TGP and NSGP fail to capture the multimodality when prediction locations are near the true boundaries.

We have also investigated the case where the data is generated from anisotropic processes. Overall, the findings are consistent with the isotropic case. See Supplementary Section B.3.3 for details.

3.6 Real Data Analysis

We apply the STGP regression model to analyze the precipitation data over the contiguous United States (CONUS)[*]. The data set consists of daily average precipitation over the 2018 water year (October 1, 2017 to September 30, 2018) obtained from the Global Histori-

[*] The data set is publicly available at `https://sites.google.com/site/markdrisser/data-sets?authuser=0`.

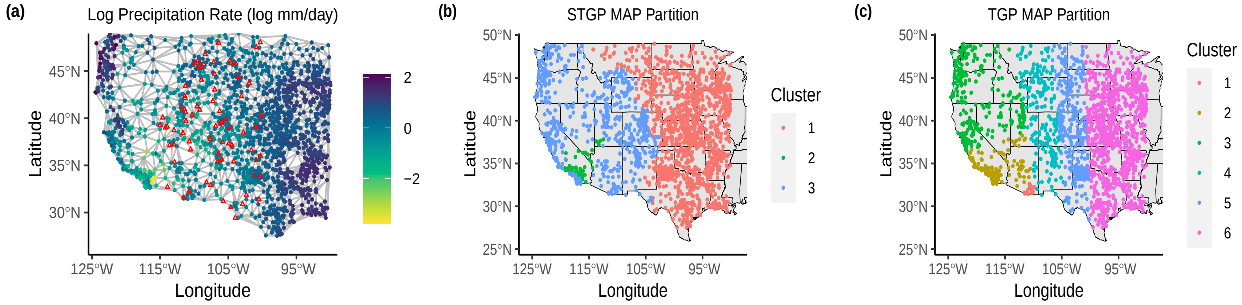

Figure 3.4: (a) Log precipitation rate measured at $n = 1689$ GHCN-D stations and the Delaunay triangulation graph used for model fitting. $r = 75$ hold-out locations near the Rocky Mountains are marked as red triangles. (b, c) MAP partition estimates of the training locations given by STGP and TGP.

cal Climatology Network-Daily database (GHCN-D), and was analyzed in Risser and Turek (2020). As noted in Risser and Turek (2020), the precipitation data in the western half of the CONUS is highly nonstationary due to the heterogeneous topography and the diverse physical phenomena related to precipitation. As a result, we focus on the precipitation data measured at GHCN-D stations located to the west of 90°W and use $n = 1689$ uniformly selected locations out of 1939 stations for model fitting. We perform a logarithmic transform of the precipitation rates following Risser and Turek (2020) so that the GP assumption is more applicable. The observed locations and the associated log precipitation rates are shown in Figure 3.4(a). The goal of this analysis is to demonstrate how well the STGP model recovers the local stationarity structure in the precipitation data and predicts the precipitation at unobserved locations, especially around boundaries.

To model the log precipitation rates, we consider a STGP regression with a spatially constant mean function (i.e., a spatially constant intercept) and a geometric anisotropic Matérn covariance function. As in the simulation studies, we compare the STGP model with TGP and NSGP. The detailed specifications of all models can be found in Supplementary Section B.4. We also perform predictive analysis of the log precipitation rates at the hold-out locations in the same manner as in Section 3.5.

Figure 3.4(b, c) shows the MAP estimates for partitions from STGP and TGP. The partition given by STGP can be largely explained by the topography in the CONUS: Cluster 1 covers the Interior Plains and the Interior Highlands to the east of the Rocky Mountains, while Cluster 3 corresponds to the mountainous regions including the Rocky Mountain System, the Intermontane Plateaus, and most parts of the Pacific Mountains. The small Cluster 2 mainly consists of the dessert region in southern California with low precipitation rates. This suggests that the STGP model can capture the geographic heterogeneity in the precipitation data. The TGP model identifies more clusters, some of which partly overlap with the clusters from STGP but others are quite different. For example, in the partition from STGP, the northern Montana region shares the same cluster membership as the regions to its east, while this is not the case in the one obtained from TGP. One possible reason is that binary trees used in TGP may partition an irregularly shaped region into several subregions with horizontal or vertical boundaries. Another possible reason is that the TGP model uses a less flexible separable exponential covariance function compared with the geometric anisotropic one in STGP.

As in the simulation studies, we use MSPE, average CRPS, and average LogS to quantify the performance of predicting out-of-sample log precipitation rates. Table 3.2 summarizes the results based on $r_1 = 75$ hold-out locations between 100°W and 115°W near the Rocky Mountains that contain many boundary points identified by STGP and TGP. The STGP models achieve the best predictive performance in all three metrics. We have also investigated the prediction results based on $r_2 = 175$ hold-out locations that are not near the Rocky Mountains area, which suggest comparable performance of all models under this prediction scenario. The details are provided in Supplementary Section B.4. In summary, our results indicate that the gain in the prediction performance when using STGP over other methods is more prominent for boundary locations.

We have also examined the predictive surfaces and SDs at equally spaced points. The results from all three models look similar and are provided in Supplementary Section B.4.

Table 3.2: Prediction performance for the precipitation data on $r_1 = 75$ hold-out locations.

	STGP ($L = 1$)	STGP ($L = 3$)	STGP ($L = 5$)	TGP	NSGP
MSPE	**0.073**	**0.073**	0.075	0.093	0.081
Mean CRPS	0.145	**0.143**	**0.143**	0.159	0.152
Mean LogS	-0.001	-0.017	**-0.019**	0.076	0.083

3.7 Conclusions and Discussion

In this chapter, we have developed a novel soft partitioned Gaussian process to capture local stationarity structures. Our process is based on a soft partition process on the spatial domain. We complement this process model with a flexible partition prior based on a predictive random spanning tree model and embed it into a Bayesian hierarchical spatial modeling framework, leading to the spanning-treed Gaussian process model. The prediction of STGP utilizes a mixture of L Gaussian distributions, where L is the number of nearest neighbors used for determining cluster memberships. A systematic way for choosing L using methods such as model selection criteria is under investigation.

Although in this work we only focus on univariate GPs, the proposed general modeling framework can be extended along several directions. It is straightforward to embed SPGP in a spatial GLM framework for the analysis of non-Gaussian spatial responses. Another future research direction is to extend the univariate process into multivariate cases, possibly with multiple spanning-treed partitions and tree based graphical models (Gao et al., 2021). Extension to soft partitioned versions of other types of stochastic processes is also possible if the conditional distribution is available. Finally, it is known that nearest neighbor graphs and Delaunay triangular meshes are capable of capturing more complex geometries. Therefore, a promising direction of future research is to extend our graph-based SPGP to build locally stationary processes on complex domains.

On the computational side, we have demonstrated that scalability can be straightforwardly achieved by specifying a small-sized set of reference knots. The ideal choice of reference knots may depend on the true but unknown partition. A possible way to achieve this is to treat the choice of reference knots as random so that the distribution of knots can be learned from data and adapt to the true partition.

Our theoretical results on the STGP models suggest that the posterior distribution of the conditional density concentrates in a weak or total variation neighborhood, and we establish a contraction rate for the latter case. We remark that the rate can be potentially improved if the complexity of the spanning-treed partition space can be better bounded. For linear prediction (or kriging) problems, posterior asymptotic efficiency can possibly be established following a similar spirit of Li (2020). We leave these as future works.

4. BAST: BAYESIAN ADDITIVE REGRESSION SPANNING TREES FOR COMPLEX CONSTRAINED DOMAIN *

4.1 Introduction

Over the past few decades, data collected from complex constrained domains have attracted much attention in machine learning and spatial statistics. Domains with non-trivial geometries, such as irregular boundaries, sharp concavities, and/or interior holes due to geographic constraints (e.g., lakes and coasts), impose challenges on statistical modeling, as the Euclidean assumption underpinning many traditional statistical and machine learning methods no longer holds for data with intrinsic geometries.

In this chapter, we consider nonparametric regression problems with features lying on constrained domains or, more generally, compact Riemannian manifolds. To be more specific, we model the response variable $Y(\mathbf{s}) \in \mathbb{R}$ at a location $\mathbf{s}$ on a compact Riemannian manifold $\mathcal{M}$ as

$$Y(\mathbf{s}) = f(\mathbf{s}) + \epsilon(\mathbf{s}), \quad \epsilon(\mathbf{s}) \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2), \tag{4.1}$$

for some unknown function $f : \mathcal{M} \to \mathbb{R}$ and noise variance σ^2. In many applications, the true function $f(\cdot)$ may not be globally smooth but has discontinuities/abrupt changes across some narrow boundary regions in the domain. For example, housing prices can be substantially different in two neighboring communities, and ocean chlorophyll data that are separated by a narrow peninsula can exhibit distinct spatial patterns. It is of great need to develop new methodologies that can both respect intrinsic geometries of the domain and capture complicated local discontinuity patterns in the true function.

There is growing literature on nonparametric regression and smoothing for complex domains. Spline smoothing methods (Ramsay, 2002; Lai and Schumaker, 2007; Wang and

[*] Reprinted from Luo, Z. T., Sang, H., and Mallick, B. (2021a). BAST: Bayesian Additive Regression Spanning Trees for Complex Constrained Domain. *Advances in Neural Information Processing Systems*, 34. The authors hold the copyright.

Ranalli, 2007; Wood et al., 2008; Scott-Hayward et al., 2014) have been developed for data on constrained domains, but most of them focus on domains in $\mathbb{R}^2$. Sangalli et al. (2013) generalized the spline methods to constrained regions in $\mathbb{R}^3$. For more general Riemannian manifolds, kernel based smoothing models, including kernel regressions (Pelletier, 2006; Henry and Rodriguez, 2009) and local regressions (Aswani et al., 2011; Cheng and Wu, 2013; Di Marzio et al., 2014), were developed. Gaussian process (GP) regression is another popular tool for nonparametric regression problems, and many works focus on developing valid covariance kernels on spheres (see Jeong and Jun, 2015; Guinness and Fuentes, 2016; Guella et al., 2018, among others). More recently, a few practical GP models for constrained domains and Riemannian manifolds were studied in the literature (Lin et al., 2019; Niu et al., 2019; Borovitskiy et al., 2020; Dunson et al., 2020). However, most of the aforementioned approaches assume globally smooth true functions and thus may not fully adapt to the ones with local discontinuities.

Ensemble tree models (Breiman, 2001; Chen and Guestrin, 2016) have been widely used in traditional nonparametric regression problems. One prominent example is the Bayesian additive regression trees (BART) model (Chipman et al., 2010), due to its versatility and capability of producing uncertainty measures. However, to our knowledge, ensemble methods have not been used for nonparametric regression on complex constrained domains, and almost all ensemble tree methods rely on binary decision tree partition models as their ensemble members (weak learners). Nevertheless, binary trees may not be ideal to capture possibly highly irregular partitions on complex domains as they can only make splits parallel to Euclidean coordinate axes. For instance, in the U-shape domain shown in Figure 4.1(c), a complicated and over-clustered binary treed partition is needed to approximate a simple partition with three clusters (marked by different colors). Moreover, the rectangular partitions do not comply to irregular domain constraints, which may cause the so called "leakage" problems on complex domains. With a similar partitioning idea, Menafoglio et al. (2018) proposed a Voronoi tessellation based model to account for domain constraints, but their method imposes convexity restrictions on partitions. Most recently, spanning treed partition models have been demonstrated as an effective tool for characterizing partitions with flexible shapes (Li and Sang, 2019; Teixeira et al., 2019; Luo et al., 2021b), but their focus is on traditional two-dimensional Euclidean spaces and linear regression settings.

Our contribution in this chapter is to propose a novel Bayesian additive regression spanning trees (BAST) model for nonparametric regressions on complex constrained domains with efficient Bayesian inference algorithms. The backbone of BAST is a new random spanning tree (RST) manifold partition model, which replaces binary decision trees in each weak learner. RST is capable of capturing irregularly shaped partitions with a small number of spanning tree edge cuts while respecting intrinsic geometries and domain boundary constraints. Equipped with a *soft* prediction scheme, we show that BAST achieves a superior prediction performance over other competing methods on various tasks, thanks to its strong local adaptivity to different levels of smoothness.

The rest of the chapter proceeds as follows. In Section 4.2, we present the RST partition models on manifolds and develop a new Bayesian nonparametric regression model with RST ensembles. Section 4.3 discusses algorithms for Bayesian inference. In Section 4.4, we illustrate the model performance by simulation experiments and a real chlorophyll data set in Aral Sea. Section 4.5 concludes the chapter with some discussions. Additional details on Bayesian inference, hyperparameter selection, and sensitivity analysis are provided in Supplementary Materials. The code of BAST is available at `https://github.com/ztluostat/BAST`.

4.2 Bayesian Nonparametric Regressions with Additive Spanning Trees

4.2.1 A Spanning Treed Partition Model on Manifolds

Our novel nonparametric regression model is built upon an ensemble of partitions of observations. In this subsection, we introduce a stochastic partition model for data on a compact Riemannian manifold via random spanning trees (RSTs), which will serve as a building block to develop the sum-of-spanning-trees model in Section 4.2.2.

Let $\mathcal{M}$ be a d-dimensional compact Riemannian manifold that is known *a priori*, and $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subseteq \mathcal{M}$ be a finite set of locations on $\mathcal{M}$ where the data are observed. We are interested in partitioning $\mathcal{S}$ into several disjoint subsets such that each subset consists of nearby locations where the data are relatively homogeneous that can be modeled separately. For spatial data, it is often desired to impose contiguity constraints on partitions. Below, we introduce the notion of spatially contiguous partitions on a manifold based on a spatial graph whose edges encode the relationship of spatial adjacency or neighborhood. Let $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ be a connected undirected graph with vertex set $\mathcal{S}$ and edge set $\mathcal{E}$ that connects $\mathbf{s} \in \mathcal{S}$ to its "close neighbors". The construction of spatial graphs on a manifold will be discussed later in this subsection. We say $\pi(\mathcal{S}) = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$, where $\mathcal{S}_j \subseteq \mathcal{S}$ for $j = 1, \ldots, k$, is a *spatially contiguous partition* of $\mathcal{S}$ with respect to $\mathcal{G}$ if $\cup_{j=1}^{k} \mathcal{S}_j = \mathcal{S}$, $\mathcal{S}_j \cap \mathcal{S}_{j'} = \emptyset$ for all $j \neq j'$, and there exists a connected subgraph $\mathcal{G}_j = (\mathcal{S}_j, \mathcal{E}_j)$ of $\mathcal{G}$ for each j. We call each $\mathcal{S}_j$ a cluster, which consists of locations that are connected to each other and thus is spatially contiguous with respect to the spatial graph. Henceforth, when there is no risk of confusion, we will refer to spatially contiguous partitions simply as partitions. Figure 4.1(a) shows an example of a partition with three clusters.

For constrained domains in $\mathbb{R}^2$ such as the U-shaped domain in Figure 4.1, spatial graphs can be constructed via constrained Delaunay triangulations (CDTs; Chew, 1989). Specifically, let $\mathcal{G}_0$ be a CDT mesh on $\mathcal{S} \cup \mathcal{S}_B$, where $\mathcal{S}_B$ is a set of locations on the domain boundaries. Then the induced subgraph of $\mathcal{G}_0$ on $\mathcal{S}$ can be chosen as a spatial graph $\mathcal{G}$. Edges longer than a certain threshold can be removed if desired. See Figure 4.1(a) for an example of $\mathcal{G}$ constructed via CDT. For a general manifold, motivated by the nice adaptive properties of nonparametric regressions based on K nearest neighbor (K-NN) graphs on manifolds (Kpotufe, 2011; Madrid Padilla et al., 2020), one may construct $\mathcal{G}$ by a K-NN graph that connects $\mathbf{s} \in \mathcal{S}$ to its K nearest neighbours with respect to geodesic distance.

Given $\mathcal{G}$, we model partitions on manifolds in a similar spirit as the spanning treed partition models developed for two-dimensional Euclidean spaces (Li and Sang, 2019; Teixeira

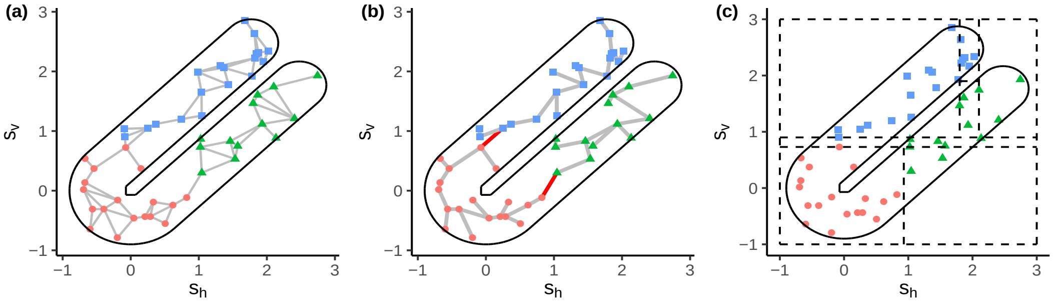

Figure 4.1: (a) A constrained Delaunay triangulation graph on a U-shaped domain. (b) A partition with three clusters obtained by removing the red edges in a spanning tree. (c) A binary treed partition nested in the three-cluster partition in (a, b).

et al., 2019; Luo et al., 2021b). Specifically, a connected subgraph $\mathcal{T} = (\mathcal{S}, \mathcal{E}_{\mathcal{T}})$ of $\mathcal{G}$ is called a spanning tree of $\mathcal{G}$ if it has no cycle. A well-known property of spanning trees is that if a set of $k-1$ edges in $\mathcal{E}_{\mathcal{T}}$ is removed, we obtain a disconnected subgraph of $\mathcal{T}$ with k connected components, which naturally defines a partition $\pi(\mathcal{S})$ with k clusters by letting $\mathcal{S}_j$ be the vertex set of the jth component. In this case, we say $\pi(\mathcal{S})$ is *induced* by $\mathcal{T}$. See Figure 4.1(b) for an example. This property implies that we can simplify a complicated graph partition modeling problem to modeling spanning trees as well as the number and locations of removed edges.

Mathematically, conditional on $\mathcal{T}$ and k we assume a discrete uniform distribution on all possible partitions induced by $\mathcal{T}$:

$$p\left\{\pi(\mathcal{S}) \mid k, \mathcal{T}\right\} \propto \mathbb{1}\left\{\pi(\mathcal{S}) \text{ is induced by } \mathcal{T} \text{ and has } k \text{ clusters}\right\}, \tag{4.2}$$

where $\mathbb{1}(\cdot)$ is an indicator function.

Next, we consider a probabilistic model on the spanning tree space. Let ω_e be the weight for an edge $e \in \mathcal{E}$ and $\boldsymbol{\omega} = \{\omega_e : e \in \mathcal{E}\}$. We assume an iid uniform distribution on the edge weights and let $\mathcal{T}$ be the resulting minimum spanning tree (MST), i.e., the spanning tree with minimum $\sum_{e\in\mathcal{E}_\mathcal{T}} \omega_e$:

$$\mathcal{T} = \text{MST}(\boldsymbol{\omega}), \quad \omega_e \overset{\text{iid}}{\sim} \text{Unif}\,(0,1), \tag{4.3}$$

where MST($\boldsymbol{\omega}$) denotes an MST of $\mathcal{G}$ based on the edge weights $\boldsymbol{\omega}$. Note that we are not assuming a discrete uniform distribution on the spanning tree space, with which it is challenging to sample spanning trees for Bayesian inference. As we will show in Section 4.3.1, our model specification leads to an exact and fast spanning tree sampler, taking advantage of the Prim's algorithm for MST constructions.

Finally, we assume a truncated Poisson distribution with mean parameter λ_k on the number of clusters:

$$k \sim \text{Poisson}(\lambda_k) \cdot \mathbb{1}(1 \leq k \leq \bar{k}), \tag{4.4}$$

where $\bar{k}$ is the pre-specified maximum number of clusters.

The following proposition states that the support of RST is rich enough to accommodate all possible spatially contiguous partitions on a manifold with no more than $\bar{k}$ clusters. The proof is postponed to Appendix C. Note that similar results do not hold for binary treed partition models. See Figure 4.1(c) for a counterexample where there does not exist a rectangle containing all the blue points without including any green or red ones.

Proposition 4.1. *Let $\pi(\mathcal{S}) = \{\mathcal{S}_1, \dots, \mathcal{S}_k\}$ be an arbitrary spatially contiguous partition. Then $\pi(\mathcal{S})$ is within the support of the partition model defined by* (4.2), (4.3), *and* (4.4) *if $k \leq \bar{k}$.*

4.2.2 A Sum-of-spanning-trees Regression Model

Given the data $\{Y(\mathbf{s}_i), \mathbf{s}_i\}_{i=1}^n$, we consider the nonparametric regression problem (4.1). Instead of assuming global continuity, we assume $f(\cdot)$ belongs to a broad class of piecewise smooth functions. We propose to model $f(\cdot)$ using a summation of weak learners based on the flexible RST partitions.

Given a partition $\pi(\mathcal{S})$ induced by $\mathcal{T}$ with k clusters and cluster-wise constants $\boldsymbol{\mu} = (\mu_1, \dots, \mu_k) \in \mathbb{R}^k$, we define a mapping from $\mathcal{S}$ to $\mathbb{R}$ as

$$g(\mathbf{s}|\pi, \mathcal{T}, k, \boldsymbol{\mu}) = \mu_j \quad \text{if } \mathbf{s} \in \mathcal{S}_j,$$

where we write $\pi = \pi(\mathcal{S})$ for conciseness. This piecewise constant function on $\mathcal{S}$ serves as a weak learner for $f(\cdot)$, which approximates $f(\cdot)$ by μ_j locally in $\mathcal{S}_j$. A Bayesian additive spanning trees (BAST) model is a summation of piecewise constant functions based on various spanning-treed partitions. Specifically, for a pre-specified $M \in \mathbb{N}$, BAST models $f(\cdot)$ as

$$f(\mathbf{s}) = \sum_{m=1}^{M} g(\mathbf{s}|\pi_m, \mathcal{T}_m, k_m, \boldsymbol{\mu}_m), \quad \mathbf{s} \in \mathcal{S}, \tag{4.5}$$

where $\pi_m = \pi_m(\mathcal{S}) = \{\mathcal{S}_1^m, \dots, \mathcal{S}_{k_m}^m\}$ is a partition with k_m clusters induced by a spanning tree $\mathcal{T}_m$ of $\mathcal{G}$ and $\boldsymbol{\mu}_m = (\mu_{m1}, \dots, \mu_{mk_m})$. Although in principle the spatial graphs for each weak learners need not be identical, we focus on the case where they share a common $\mathcal{G}$ for simplicity.

For $\mathbf{s} \in \mathcal{S}$, the additive structure (4.5) implies that $f(\mathbf{s})$ equals the summation of the μ_{mj}'s corresponding to the clusters from each weak learner that $\mathbf{s}$ lies in. Figure 4.2 illustrates a summation of two spanning treed partitions. This together with the shrinkage priors to be discussed in Section 4.2.3 allows each weak learner to explain a small amount of the variation in the response variable. The step function approximation also allows for capturing both smoothness and discontinuities/abrupt changes in $f(\cdot)$ (Ročková and van der Pas, 2020). In particular, our model is more efficient than some existing ensemble binary tree and smoothing methods in recovering irregularly shaped regions where discontinuities happen, thanks to the versatility of RST in capturing highly flexible cluster shapes for complex constrained domains.

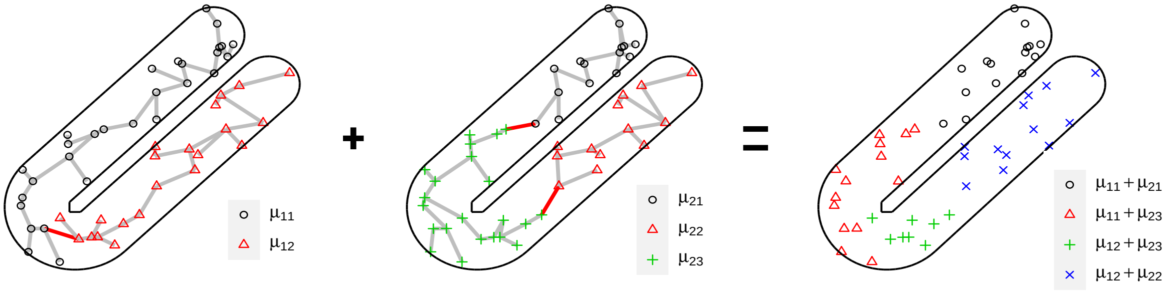

Figure 4.2: Demonstration of the partition obtained by adding two spanning treed partitions.

4.2.3 Prior Regularization

Similar to BART (Chipman et al., 2010), shrinkage priors play an important role in regularizing weak learners and preventing overfitting for BAST. In this subsection we discuss the specification of prior models, which admits the form

$$p\left(\{\pi_m, \mathcal{T}_m, k_m, \boldsymbol{\mu}_m\}_{m=1}^M, \sigma^2\right) = \left\{\prod_{m=1}^M p(\boldsymbol{\mu}_m|\pi_m, \mathcal{T}_m, k_m) p(\pi_m, \mathcal{T}_m, k_m)\right\} p(\sigma^2).$$

We assign an iid RST prior for $(\pi_m, \mathcal{T}_m, k_m)$ given by (4.2), (4.3), and (4.4). Small values of λ_k and $\bar{k}$ in (4.4) are typically chosen to restrict the number of clusters in each partition, leading to simpler piecewise constant structures in each weak learner which prevent overfitting and encourage better mixing in Markov chain Monte Carlo. Note that we regularize the number of clusters in a more direct way than binary treed partition priors (Chipman et al., 1998, 2010) which implicitly penalize large numbers of clusters by increasing the prior probability that a node is terminal as the depth of the node.

Rescaling $Y(\mathbf{s})$ into $[-0.5, 0.5]$, we opt to place a shrinkage prior that concentrates around 0 for $\boldsymbol{\mu}_m$ following (Chipman et al., 2010). Conditional on $(\pi_m, \mathcal{T}_m, k_m)$, we choose a conjugate prior for $\boldsymbol{\mu}_m$ given by

$$\boldsymbol{\mu}_m|\pi_m, \mathcal{T}_m, k_m \sim \mathrm{N}_{k_m}(\mathbf{0}, \sigma_\mu^2 \mathbf{I}_{k_m}), \tag{4.6}$$

independently for $m = 1, \ldots, M$, where σ_μ is assumed to depend on the number of weak learners, specifically, $\sigma_\mu = 0.5/(a\sqrt{M})$ with $a > 0$. Intuitively, when we have a larger number of weak learners, it is desired to impose stronger shrinkage effects by choosing a smaller σ_μ such that each learner is not too influential to the overall fit. A default choice of a can be $a = 2$, which assigns 0.95 prior probability for $f(\mathbf{s})$ that lies in $[-0.5, 0.5]$.

Finally, we assign a conjugate inverse-χ^2 prior on σ^2: $\sigma^2 \sim \nu\lambda_s/\chi^2_\nu$. We fix $\nu = 3$ and choose λ_s in a data-driven way such that the prior satisfies $\mathbb{P}(\sigma^2 < \hat{\sigma}^2) = 0.90$ similarly as in BART (Chipman et al., 2010), where $\hat{\sigma}^2$ is the sample variance of $\mathbf{Y} = \{Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_n)\}$.

4.3 Bayesian Inference

4.3.1 Estimation

Inference of BAST is based on a tailored backfitting Markov chain Monte Carlo (MCMC) algorithm (Hastie and Tibshirani, 2000), in which we successively sample $(\pi_1, \mathcal{T}_1, k_1, \boldsymbol{\mu}_1), \ldots, (\pi_M, \mathcal{T}_M, k_M, \boldsymbol{\mu}_M)$, and σ^2 from their respective full conditionals. Our sampler for the full conditional $p\left(\pi_m, \mathcal{T}_m, k_m, \boldsymbol{\mu}_m | -\right)$, where $-$ stands for all other parameters and the data $\mathbf{Y}$, consists of two successive steps: (i) we first analytically integrate $\boldsymbol{\mu}_m$ out and sample from the collapsed conditional distribution for the partition parameters $p(\pi_m, \mathcal{T}_m, k_m | -)$, and (ii) we then sample $\boldsymbol{\mu}_m$ from $p(\boldsymbol{\mu}_m | \pi_m, \mathcal{T}_m, k_m, -)$. This design leads to better mixing and convergence performance of the sampler by avoiding the trans-dimensional problem for $\boldsymbol{\mu}_m$'s. Thanks to the conjugate priors, sampling from $p(\boldsymbol{\mu}_m | \pi_m, \mathcal{T}_m, k_m, -)$ and $p(\sigma^2 | -)$ follows standard procedures, and we leave the details to Appendix A.1.

Below, we focus on the sampling of the RST partition parameters. To draw samples of $(\pi_m, \mathcal{T}_m, k_m)$, one of the four moves — *birth*, *death*, *change*, and *hyper* — is performed with probabilities $r_b(k_m)$, $r_d(k_m)$, $r_c(k_m)$, and $r_h(k_m)$, respectively (Luo et al., 2021b). The first three moves modify the partition by proposing a new partition induced by the current spanning tree, and the hyper move updates $\mathcal{T}_m$ by sampling from its full conditional via an efficient sampling algorithm. Each move is detailed below and some of them are visualized

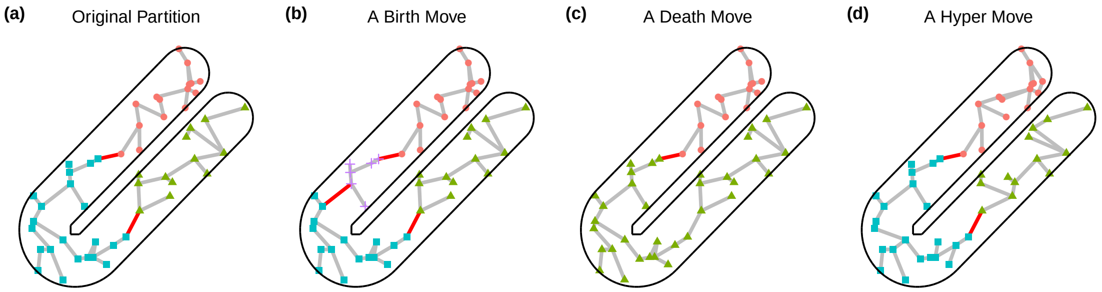

Figure 4.3: Partitions and spanning trees obtained after (b) a birth, (c) a death, or (d) a hyper move from the original partition and tree in (a).

in Figure 4.3.

In a birth move, one of the clusters is split into two by randomly removing an edge in $\mathcal{T}_m$ that connects vertices belonging to the same cluster. Denoting the new partition by π_m^*, the Metropolis-Hastings (M-H) acceptance ratio is given by

$$\min\left\{1,\ \frac{\lambda}{(k_m+1)} \times \frac{r_d(k_m+1)}{r_b(k_m)} \times \frac{\mathcal{L}\left(\mathbf{Y}|\pi_m^*, \mathcal{T}_m, k_m+1, -\right)}{\mathcal{L}\left(\mathbf{Y}|\pi_m, \mathcal{T}_m, k_m, -\right)}\right\}, \tag{4.7}$$

where $\mathcal{L}\left(\mathbf{Y}|\pi_m, \mathcal{T}_m, k_m, -\right)$ is the integrated likelihood with $\boldsymbol{\mu}_m$ marginalized out, whose closed form can be found in Appendix A.1. Opposite to the birth move, a death move randomly merges two adjacent clusters in π_m. Specifically, an edge in $\mathcal{T}_m$ that connects two distinct clusters in π_m is uniformly selected and then the two clusters are merged into one. The M-H ratio is analogous to (4.7). In a change move, a death move is performed followed by a birth move, so that the number of clusters is unchanged. This move is designed to encourage better mixing of the sampler.

Finally, a hyper move updates $\mathcal{T}_m$ using an exact sampler, which adaptively learns a spanning treed spatial order so that we can obtain a partition that is more compatible to the homogeneity pattern of data in subsequent MCMC iterations. Specifically, we sample the edge weight ω_e of $\mathcal{G}$ from iid $\text{Unif}(1/2, 1)$ if two endpoints of e are in different clusters under π_m, and otherwise from iid $\text{Unif}(0, 1/2)$. A new spanning tree is the MST generated by Prim's algorithm using the new edge weights. It can be shown that the resulting MST induces the current partition (Teixeira et al., 2019) and is an exact sample from its full conditional distribution (Luo et al., 2021b).

The overall computational complexity per MCMC iteration is $O\big(M((1-r_h)n+r_h n\log n)\big)$, where r_h is the probability that a hyper step is selected which takes $O(n\log n)$ using Prim's algorithm for CDT and K-NN graphs, and $O(n)$ is the computation complexity required when birth/death/change steps are selected because a closed form marginal likelihood without matrix inversion is available when calculating acceptance ratios. In practice, we suggest a small value of r_h such as 0.1 to reduce the computation and allow the algorithm to spend more iterations on learning a good partition compatible with the current tree. To further reduce computation complexity, we have done some preliminary exploration of using *different* but *fixed* spanning trees for each weak learner during MCMC (i.e., setting $r_h = 0$). As shown in Appendix B.1.3, this significantly speeds up the computation while the prediction performance remains comparable.

4.3.2 Prediction

The prediction at an unobserved location $\mathbf{u} \notin \mathcal{S}$ involves two steps. First, in each weak learner, we randomly assign $\mathbf{u}$ to one of its nearby clusters subject to the manifold constraints, using a *soft* prediction scheme in a similar spirit to Linero and Yang (2018). Second, the prediction is obtained by summing the constants corresponding to the clusters that $\mathbf{u}$ belongs to over all weak learners.

Specifically, to obtain cluster memberships for $\mathbf{u}$, we define its neighbor set $N_{\mathbf{u}} \subseteq \mathcal{S}$ as follows. For a constrained domain in $\mathbb{R}^2$, $N_{\mathbf{u}}$ is chosen as the vertices of the triangle containing $\mathbf{u}$ in the CDT mesh that belong to $\mathcal{S}$ (i.e., vertices on the domain boundary are excluded; see Appendix A.2 for detailed discussions). For general manifolds in higher dimensional spaces, $N_{\mathbf{u}}$ is specified as the set of K nearest neighbors of $\mathbf{u}$ in $\mathcal{S}$ with respect to the geodesic distance.

Let $z_m(\mathbf{v}) \in \{1, \ldots, k_m\}$ be the cluster membership of a generic location $\mathbf{v} \in \mathcal{M}$ from the

mth weak learner such that $z_m(\mathbf{s}) = j$ if $\mathbf{s} \in \mathcal{S}_j^m$. Intuitively, $\mathbf{u}$ is expected to share the same cluster membership as one of its neighbors in $\mathcal{S}$, and if $\mathbf{u}$ is near the boundary of a cluster in a partition, it is more ideal to assign $z(\mathbf{u})$ probabilistically to reflect the partitioning uncertainty and adapt for smoother functions. This motivates us to consider the following random assignment for $z_m(\mathbf{u})$'s: given $N_\mathbf{u}$ and a posterior sample of the partitions, $z_m(\mathbf{u})$ is sampled independently such that $\mathbb{P}\{z_m(\mathbf{u}) = z_m(N_{\mathbf{u},\ell})\} = \alpha_\ell$, for $\ell = 1, \dots, |N_\mathbf{u}|$, where $N_{\mathbf{u},\ell}$ is the ℓth element in $N_\mathbf{u}$ and α_ℓ satisfies $\sum_{\ell=1}^{|N_\mathbf{u}|} \alpha_\ell = 1$. One can specify α_ℓ by setting $\alpha_\ell = 1/|N_\mathbf{u}|$ for all ℓ, or via inverse geodesic distance weighting such as $\alpha_\ell \propto 1/d_g^b(\mathbf{u}, N_{\mathbf{u},\ell})$, where $d_g(\cdot,\cdot)$ is the geodesic distance and b is some positive power.

At the second step, we sum $\mu_{m,z_m(\mathbf{u})}$ over $m = 1, \dots, M$ to obtain a posterior predictive value of $\mathbb{E}\{Y(\mathbf{u})\}$ given samples of $z_m(\mathbf{u})$'s. A point predictor for $Y(\mathbf{u})$ can then be taken as the mean of the posterior draws, which allows us to average models with different RST partition structures.

Finally, we remark that the prediction algorithm is highly parallelizable, as the predictive sampling for each RST partition is independent.

4.4 Experiments

4.4.1 U-shape Example

We first examine the BAST's performance of recovering piecewise smooth functions via some simulation experiments on a rotated U-shaped domain shown in Figure 4.4(a). Our true function $f(\cdot)$ is constructed based on the one in Ramsay (2002), denoted as $f_R(\cdot)$. We create discontinuities along a circle of radius 0.9 centered at the origin as follows. For locations inside the circle, we flip f_R by setting $f = -f_R$. For locations outside the circle, we set $f = 2f_R$ for those in the lower arm of the domain, and $f = f_R$ for those in the upper arm, such that the jump in the lower arm has a larger magnitude. We uniformly generate $n = 500$ locations in the domain as training data and 200 out-of-sample locations for prediction. Figure 4.4(a) shows the true function in the training data. The responses are generated according to (4.1) with different levels of noise $\sigma = 0.1, 0.5, 1$, and each noise level is replicated for 50 times.

The spatial graph $\mathcal{G}$ is constructed via CDT. We use $M = 20$ weak learners and set $\lambda_k = 4$ and $\bar{k} = 10$ to restrict the size of each partition. The prediction is based on the CDT graph, and we use inverse distance weighting with $b = 1$ to sample cluster memberships. The probabilities for MCMC moves are set as $r_b = r_d = r_c = 0.3$ and $r_h = 0.1$, with adjustments for cases where $k_m = 1$ or $\bar{k}$. We compare BAST with BART (Chipman et al., 2010), and two other nonparametric regression methods for constrained domains, the soap film smoothing (SFS; Wood et al., 2008) and the sparse intrinsic Gaussian process (inGP) regression (Niu et al., 2019). For BART, we use the same number of weak learners as in BAST and use its default settings. We run the MCMC for both BAST and BART for $20,000$ iterations, discarding the first half and retaining samples every 5 iterations. Our MCMC diagnostics suggest no convergence issue of BAST. We specify 32 equally spaced knots for SFS and set its basis dimension as 40. For sparse inGP, we use 24 equally spaced knots and simulate Brownian motions for $100,000$ times. The prediction performance over 200 out-of-sample testing locations is assessed by mean squared prediction errors (MSPEs) and mean absolute prediction errors (MAPEs). For the two Bayesian approaches BAST and BART, we also compare their probabilistic prediction performance gauged by continuous ranked probability scores (CRPSs) based on their posterior predictive distributions (see, e.g., Gneiting and Raftery, 2007). For all the metrics, lower values indicate better performance.

Table 4.1 summarizes the average performance metrics over 50 replicates for each noise level. BAST outperforms all its competitors in terms of all the metrics under all the noise levels. This is because BAST partitions the training data in a way that adapts to both the domain constraints and the irregularly shaped discontinuity boundaries, thanks to the flexible RST partitions. In contrast, the binary treed partitions adapt to neither types of boundaries, and neither of SPS and inGP captures discontinuities in the true function. This is also evidenced by Figure 4.4(b-e), where absolute prediction errors (APEs) of the test data in one

Table 4.1: Prediction performance of BAST and its competing methods in the U-shape domain example. Standard errors are given in parentheses.

		BAST	BART	SFS	inGP
$\sigma = 0.1$	MSPE	**0.189** (0.001)	1.541 (0.075)	0.418 (0.001)	0.814 (0.002)
	MAPE	**0.188** (0.001)	0.436 (0.010)	0.340 (0.001)	0.610 (0.001)
	Mean CRPS	**0.142** (0.001)	0.380 (0.009)	—	—
$\sigma = 0.5$	MSPE	**0.464** (0.006)	1.704 (0.053)	0.680 (0.007)	1.057 (0.010)
	MAPE	**0.491** (0.004)	0.686 (0.008)	0.591 (0.004)	0.752 (0.004)
	Mean CRPS	**0.371** (0.003)	0.575 (0.007)	—	—
$\sigma = 1$	MSPE	**1.283** (0.018)	2.650 (0.056)	1.491 (0.020)	1.823 (0.025)
	MAPE	**0.888** (0.007)	1.072 (0.008)	0.951 (0.007)	1.051 (0.008)
	Mean CRPS	**0.693** (0.006)	0.889 (0.008)	—	—

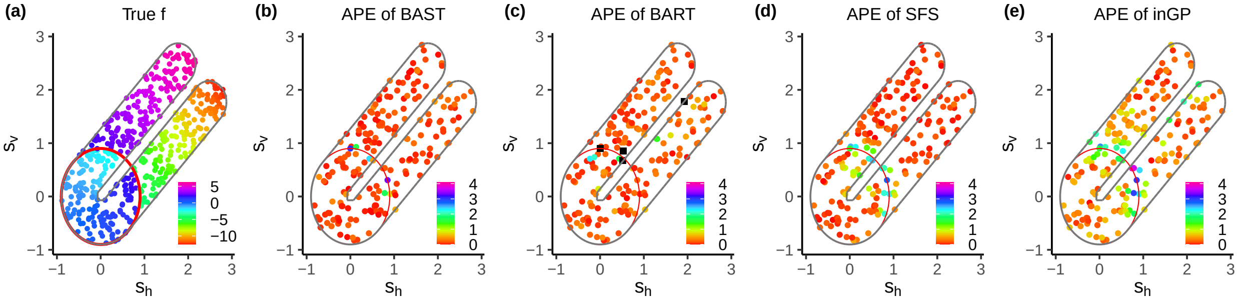

Figure 4.4: (a) True function in the training data for the U-shape domain example. (b-e) APEs of one test data set with $\sigma = 0.1$. The discontinuity boundaries are marked as red circles. Black squares in (c) indicate APE > 4.10.

replicate with $\sigma = 0.1$ are shown. The APEs from BAST are small for most locations except for those near the discontinuity boundaries. SFS also has similar patterns; however, errors near the discontinuity boundaries are much higher due to the global smoothness assumption in SFS. The general APE pattern for inGP is similar to SFS, except that it has larger errors, possibly due to the low-rank approximation of covariance functions. With the same number of weak learners, BART has larger errors near the upper discontinuity boundary, as more rectangular partitions are needed to well approximate irregular boundaries. Moreover, it also gives poor prediction at some locations near the domain boundary between the two arms, probably because binary treed partitions do not take into account the domain boundary when making axis parallel splits, and hence force some boundary locations in one arm to share the same cluster memberships with those locations in the other arm. We have also experimented using more weak learners in BART, and the results in Appendix B.1.1 suggest that BAST with a fewer number of weak learners still outperforms BART. Computation time for each method is reported and compared in Appendix B.1.3.

Hyperparameters of BAST can be tuned using standard cross-validation techniques. Our results in Appendix B.1.2 show that the fine-tuned BAST with respect to M, $\bar{k}$ and the shrinkage parameter a for σ_μ achieves better performance than the default version in Table 4.1, but the performance of them is close to each other. We have also conducted additional sensitivity analyses to the hyperparameters M, $\bar{k}$, and λ_k in Appendix B.1.2, which suggests that the performance of BAST is in general robust to them.

4.4.2 Bitten Torus Example

To illustrate BAST for more general manifolds, we consider a bitten torus example similar to Niu et al. (2019). A torus is a two-dimensional manifold embedded in $\mathbb{R}^3$ that is parameterized by (θ, ϕ), where θ is the angle for the torus and ϕ is the angle for the tube. Let R be the fixed distance from the center of the tube to the center of the torus, and r be the fixed radius of the tube. The Cartesian coordinate (x, y, z) on a torus can be written as $x = (R + r\cos\theta)\cos\phi$, $y = (R + r\cos\theta)\sin\phi$, and $z = r\sin\theta$. We create a bitten torus by

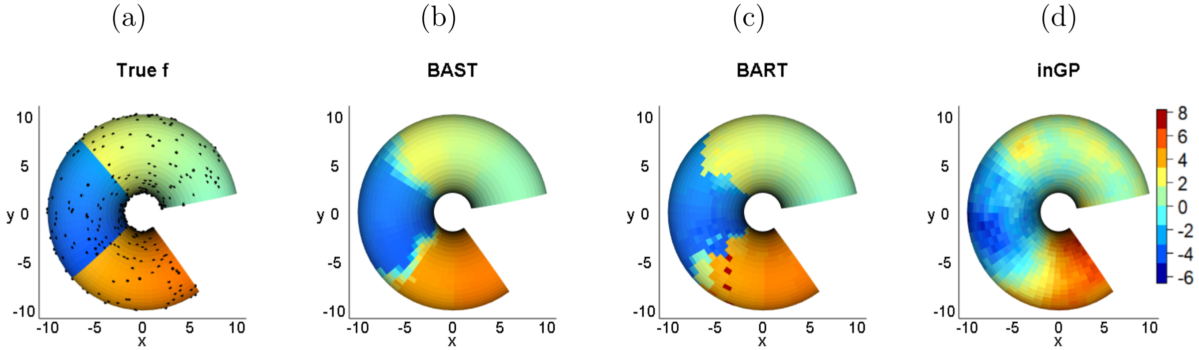

Figure 4.5: (a) True function and training locations (marked as black dots). (b-d) Predictive surfaces of BAST and its competing methods. All plots are viewed along the negative direction of the z-axis.

setting $\phi \in [\pi/6, 1.7\pi]$ and $\theta \in [0, 2\pi]$.

We consider a piecewise smooth true function $f(x, y, z)$ defined on the bitten torus. We divide the torus into three subregions corresponding to $\theta \in [\pi/6, 3\pi/4]$, $\theta \in (3\pi/4, 5\pi/4]$, and $\theta \in (5\pi/4, 1.7\pi]$, respectively. The true functions in the first and the third regions are the same as the one used in Niu et al. (2019), while we set the one in the second region as the negative of the function in Niu et al. (2019), such that there are jumps along $\theta = 3\pi/4$ and $5\pi/4$. We generate responses at $n = 500$ random locations according to (4.1) with $\sigma = 0.1$ as training data. The true function and the training locations are shown in Figure 4.5.

We construct the spatial graph by a 10-NN graph based on the geodesic distance. Since the geodesic distance of a torus has no analytic form, we approximate it as in Isomap algorithm (Tenenbaum et al., 2000). Specifically, we first construct a weighted, Euclidean distance based nearest neighbor graph on some fine grids and the training locations. Then we approximate the geodesic distance between two training locations by the length of the shortest path between them in the graph. For prediction at an unobserved location $\mathbf{u}$, we use its 5 nearest neighbors in $\mathcal{S}$ based on the geodesic distance as its neighbor set $N_{\mathbf{u}}$. We compare BAST with BART that uses Cartesian coordinates as features and sparse inGP that uses 24 equally spaced knots, as SFS is only applicable for domains in $\mathbb{R}^2$. Other model specifications are the same as those in Section 4.4.1.

Table 4.2: Prediction performance of BAST and its competing methods in the bitten torus example. Standard errors are given in parentheses.

	BAST	BART	inGP
MSPE	**0.487** (0.002)	1.115 (0.041)	2.283 (0.005)
MAPE	**0.307** (0.001)	0.406 (0.009)	1.159 (0.003)
Mean CRPS	**0.225** (0.002)	0.355 (0.008)	—

As in the previous experiment, we compare the prediction performance at 200 random out-of-sample locations. The average performance metrics over 50 replicates in Table 4.2 suggest that BAST provides the most accurate prediction. We further compare the predictive surfaces of the three methods based on $2,500$ grid points in Figure 4.5(b-d). As expected, both BAST and BART capture the piecewise structure in the true function, whereas inGP does not. In the interior of each subregion, BAST and BART approximate the true smooth functions fairly accurately. The major difference between the two methods occurs near the discontinuity boundaries. In the surface from BART, some partition boundaries are parallel to the Euclidean coordinate axes such as those near $x = -5$ and $y = \pm 2$, due to the use of binary trees, while this pattern does not appear in the one from BAST. Overall, the blue subregion recovered by BAST is more consistent with the truth. Notice that the estimated function at the discontinuity boundaries from BAST is smoother thanks to its soft prediction scheme. We have also experimented with different noise levels in Appendix B.2, and the findings are consistent.

4.4.3 Application to Chlorophyll Data

We apply BAST to analyze average remote sensed chlorophyll data in the Aral data over 1998-2002, which are available in the `R` package `gamair` (Wood, 2006). The chlorophyll measurements at 485 equally spaced locations are shown in Figure 4.6(a). The southern part of the domain is separated by the isthmus of the peninsula near 59°E, and both shores of the peninsula have substantially different chlorophyll levels. It is thus desired to take into account this geographical constraints when modeling the data. The goal of our analysis is

Table 4.3: Prediction performance of BAST and its competing methods for the chlorophyll data.

	BAST	BART	SFS	inGP
MSPE	**2.346**	2.933	2.894	3.191
MAPE	**0.905**	1.172	1.071	1.200
Mean CRPS	**0.633**	0.955	—	—

to assess how well BAST captures the patterns of chlorophyll and predicts for unobserved locations in this complex spatial domain.

We follow Niu et al. (2019) to rescale the domain and model the chlorophyll level as a function of the scaled longitude and latitude plus some Gaussian noise. We use a same setup for BAST as in Section 4.4.1, except that we set $M = 30$ and $\bar{k} = 5$ to encourage smaller sizes of partitions as the number of weak learners is increased. For prediction, we sample the cluster membership of an out-of-sample location $\mathbf{u}$ using equal sampling probabilities $\alpha_\ell = 1/|N_{\mathbf{u}}|$. For BART, SFS, and sparse inGP, we also use the same settings as in the simulation experiments but with 30 trees for BART and 42 equally spaced knots for both SFS and inGP. The MCMC algorithms for BAST and BART are run for $30,000$ iterations, keeping samples every 5 iterations from the second half.

We first compare the prediction performance of all the models via 10-fold cross-validation. In the end, the chlorophyll level at each observed location is predicted exactly once, and we compare MSPEs and MAPEs based on all locations in the data set. Similarly, the mean CRPSs over all locations are compared between BAST and BART. Table 4.3 shows prediction performance metrics for four models. BAST achieves the best performance among all the models.

Next, we turn to the predictive surfaces from each model, which are shown in Figure 4.6(b-e). All models capture the general patterns of the data. The predictive surfaces from SFS and inGP are fairly smooth, while some sharp jumps can be observed in the one from BART. The surface from BAST is somewhat in between, and preserves small-scale spatial dependence of

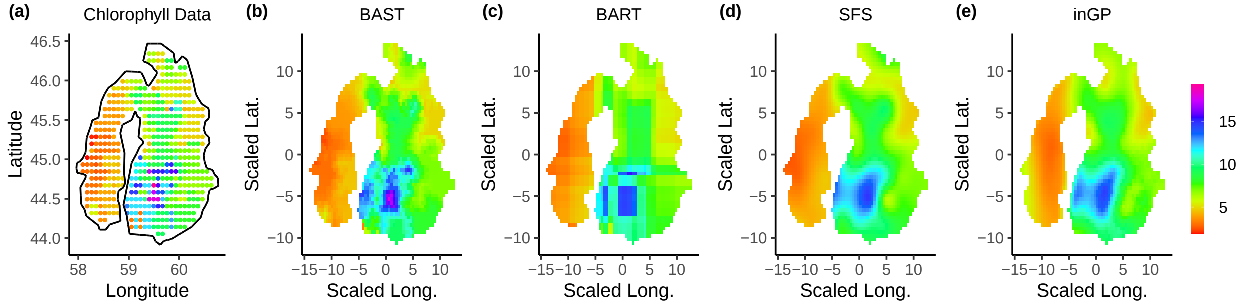

Figure 4.6: (a) Observed chlorophyll data. (b-e) Predictive surfaces from BAST and its competitors.

the data. At the southern part of the eastern basin, BART identifies a rectangular region with high chlorophyll level, while the corresponding region obtained from BAST has irregular shape, which is more consistent with the data and the results from SFS and inGP. This is due to the highly flexible RST partition model that can give irregularly shaped clusters.

4.5 Conclusion and Discussion

In this chapter, we developed a novel Bayesian nonparametric regression model on known manifolds and complex constrained domains using additive RST partitions. The RST weak learner enjoys flexibly shaped partitions while respecting the intrinsic geometries and domain constraints. The additive piecewise constant structure further allows BAST to approximate piecewise smooth functions with irregular boundaries of discontinuities, as evidenced by our simulation studies and real data analysis. In the case where the manifold is unknown, one may estimate the geodesic distance from the data (see, e.g., Meng et al., 2008, and references therein) to construct spatial graphs. We leave this scenario for future research.

Similar to its binary treed counterpart BART, BAST is promising to serve as prior models in many other Bayesian hierarchical modeling settings, such as classification models with binary and multinomial responses (Chipman et al., 2010; Murray, 2020), survival analysis (Bonato et al., 2011; Sivaganesan et al., 2017), causal inference (Hahn et al., 2020), and varying coefficient regressions (Deshpande et al., 2020). As BAST is built upon a spatial graph, it is an interesting direction to extend our methodology for classification and regression on general graphs and networks (e.g., Borovitskiy et al., 2021). Finally, theoretical justifications are important but usually challenging for ensemble methods. For example, theoretical studies of BART have begun emerging only very recently (Ročková and van der Pas, 2020; Ročková and Saha, 2019). Posterior concentration results of BAST for estimating the true function can be potentially established in similar manners as BART, but non-trivial extensions are required to theoretically handle the complex spanning tree partition on manifolds and hence beyond the scope of this work.

5. BAMDT: BAYESIAN ADDITIVE PARTIAL MULTIVARIATE DECISION TREES FOR NONPARAMETRIC REGRESSION

5.1 Introduction

In this chapter, we focus on a nonparametric regression problem with response $Y \in \mathbb{R}$ (e.g., housing price). We consider features $\mathbf{s} \in \mathcal{M}$ with *known* multivariate structures, where $\mathcal{M}$ may be a Euclidean space or a compact Riemannian manifold. For instance, $\mathbf{s}$ may represent the coordinates of a location in a spatial domain with or without boundary constraints. In addition to $\mathbf{s}$, we also consider features $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^p$ either without multivariate structures or with *unknown* multivariate structures (e.g., square footage and housing age). To be more precise, we model Y as

$$Y = f(\mathbf{s}, \mathbf{x}) + \epsilon, \quad \epsilon \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2), \tag{5.1}$$

where $f : \mathcal{D} \to \mathbb{R}$ is an unknown function defined on the joint input feature space $\mathcal{D}$, and σ^2 is an unknown noise variance. Throughout this chapter, we will refer to $\mathbf{s}$ as *structured features*, $\mathbf{x}$ as *unstructured features*, and the regression setting in (5.1) as *structured regression.*

Structured regression problems are increasingly common in many applications. Examples include spatial regressions and image analysis on complex constrained domains with nontrivial geometries, such as cities with irregular boundaries or interior holes (e.g., lakes and parks), road networks, and brain cortical surfaces, as well as prediction problems on networks using both network topology and node attributes as predictors. The general model formulation in (5.1) encompasses many classes of models as special specifications of $f(\mathbf{s}, \mathbf{x})$. Below, we focus on reviewing semi-parametric or nonparametric methods due to their flexibility in function estimation compared to parametric methods.

Related work. Spline smoothings and Gaussian process (GP) regressions are popular choices for nonparametric structured regression problems, and there have been some recent extensions of these methods for data on complex domains (Wood et al., 2008; Scott-Hayward et al., 2014; Niu et al., 2019; Borovitskiy et al., 2020; Dunson et al., 2020). However, these methods often assume globally smooth true functions and thus may not fully adapt to functions with local discontinuities. And the effects of structured features and other unstructured features are usually modeled separately. For example, conventional spatial GP regressions (Gelfand et al., 2010) add a parametric model such as a linear regression for the effects of unstructured features to a GP model for spatial effects. However, the parametric model for the regression mean part suffers the risk of being mis-specified, and the additive form could not capture the potential interactions between the effects of $\mathbf{s}$ and $\mathbf{x}$.

Alternatively, ensemble and boosting tree methods such as random forest (Breiman, 2001) and XGBoost (Chen and Guestrin, 2016) have gained great success in nonparametric prediction tasks, owning to their ability to capture both smooth and discontinuous patterns with strong local adaptivities for function estimations. In particular, Bayesian Additive Regression Trees (BART; Chipman et al., 2010) and their variants (see, e.g., Tan and Roy, 2019; He et al., 2019) offer a flexible Bayesian treatment of boosting to probabilistically model and estimate (latent) nonparametric functions in various modeling contexts, while producing uncertainty measures. These models are also appealing for handling a relatively large number of unstructured features; the decision tree weak learner often assumes a simple axis-parallel split rule based on a univariate feature at each decision node, allowing the method to more conveniently adapt to the increasing dimension of features. Nevertheless, the simple axis-parallel univariate split rule comes with a cost: the feature space can only be partitioned into (hyper) rectangular shapes which may not comply to irregular domain constraints and function discontinuity boundaries in the multivariate structured feature space. This limitation has motivated some attempts to relax the axis-parallel decision boundary assumption by considering more flexible decision split rules based on multivariate features (for review, see, e.g., Cañete-Sifuentes et al., 2021; Fan et al., 2021). However, stringent parametric assumptions such as linear or quadratic split rules (Yıldız, 2011; Blaser and Fryzlewicz, 2016) are often made in these attempts, and their estimation procedures are usually not likelihood-based and hence are lacking of uncertainty measures.

Most recently, Luo et al. (2021a) proposed a Bayesian additive model built upon random spanning tree partitions as each weak learner. However, the method is applicable to the case with structured features only. It is not straightforward to extend their method to structured regression problems with additional unstructured features, since their partition model is not formulated as decision trees. Moreover, their model is defined only for a finite number of observations. Therefore, although function estimation can be done following Bayesian inference, the out-of-sample prediction of their method is based on a two-step soft nearest neighbor approach due to the lack of a coherent Bayesian model defined on the whole manifold.

Our contributions. In light of these limitations in the current literature, we propose a new Bayesian nonparametric structured regression model for f, which is built upon an ensemble of novel partially structured multivariate decision trees (MDTs). Specifically, our decision tree recursively splits data into nodes of the tree starting from a root node. We model each node split rule by a mixture model between a *multivariate split* based on the structured feature, $\mathbf{s}$, and a *univariate split* based on one unstructured feature of $\mathbf{x}$. This allows us to combine their merits for capturing the complex effects of $\mathbf{s}$ and handling possibly high dimensional $\mathbf{x}$, and to model the interactions between $\mathbf{s}$ and $\mathbf{x}$. The multivariate split rules are built upon a novel bipartition model via predictive spanning trees. It differs from those of existing MDT methods in that: 1) it allows highly flexible decision boundary shapes while fully respecting intrinsic geometry of the structured feature space; 2) it is built on any arbitrary subset of the manifold so that both parameter estimation and prediction can be performed under a unified framework; 3) the predictive spanning tree can be constructed on a reduced dimensional reference knot set that is allowed to vary across weak learners, which can be viewed as an adaptive and multivariate extension of the binning ideas used in boosting methods such as lightGBM (Ke et al., 2017) for reduced computations.

5.2 Bayesian Structured Regression with Additive Multivariate Decision Trees

In Section 5.2.1, we introduce a new model of multivariate split rules for structured features lying on a manifold via predictive spanning tree partitions. In Section 5.2.2, we propose a novel decision tree model combining both multivariate split rules for structured features and univariate split rules for unstructured features. A Bayesian additive model of the proposed decision trees is developed in Section 5.2.3 for nonparametric structured regression problems.

5.2.1 Multivariate Splits via Predictive Spanning Tree Bipartitions

Let $\mathcal{M}$ be a known d-dimensional connected compact Riemannian manifold embedded in a Euclidean space with a geodesic distance metric d_g, and $\mathcal{S}^* = \{\mathbf{s}_1^*, \dots, \mathbf{s}_t^*\} \subseteq \mathcal{M}$ be a finite set of reference knots on $\mathcal{M}$ which may or may not coincide with the observed structured features. Typical choices of $\mathcal{S}^*$ include grid points covering $\mathcal{M}$ or a random subset of the observed values of $\mathbf{s}$. The decision tree models to be introduced in Section 5.2.2 would require a *bipartition* of certain subset $\mathcal{M}_\eta$ of $\mathcal{M}$ corresponding to a decision tree node η (see the colored region in Figure 5.1(a) for an example of $\mathcal{M}_\eta$). Let $\mathcal{S}_\eta^* \subseteq \mathcal{S}^*$ denote the union of the nearest reference knot of each point in $\mathcal{M}_\eta$ under d_g. Note that knots in $\mathcal{S}_\eta^*$ may not belong to $\mathcal{M}_\eta$. Below, we consider how to induce a *bipartition* of a generic $\mathcal{M}_\eta$ from a *bipartition* of $\mathcal{S}_\eta^*$.

For a generic set $\mathcal{A}$, we use $\pi_2(\mathcal{A}) = \{\mathcal{A}_1, \mathcal{A}_2\}$ to denote a bipartition of $\mathcal{A}$ that satisfies $\emptyset \subsetneqq \mathcal{A}_1 \subsetneqq \mathcal{A}$ and $\mathcal{A}_2 = \mathcal{A} \setminus \mathcal{A}_1$. Let $d_g(\mathbf{s}, \mathcal{B}) := \inf_{\mathbf{t}\in\mathcal{B}} d_g(\mathbf{s}, \mathbf{t})$ be the distance between $\mathbf{s}$ and a non-empty subset $\mathcal{B}$ of $\mathcal{M}$. Given $\pi_2(\mathcal{S}_\eta^*) = \{\mathcal{S}_{\eta,1}^*, \mathcal{S}_{\eta,2}^*\}$, $\pi_2(\mathcal{M}_\eta) = \{\mathcal{M}_{\eta,1}, \mathcal{M}_{\eta,2}\}$ can be obtained by setting

$$\mathcal{M}_{\eta,1} = \{\mathbf{s} \in \mathcal{M}_\eta : d_g(\mathbf{s}, \mathcal{S}_{\eta,1}^*) \le d_g(\mathbf{s}, \mathcal{S}_{\eta,2}^*)\}, \tag{5.2}$$

$$\mathcal{M}_{\eta,2} = \mathcal{M}_\eta \setminus \mathcal{M}_{\eta,1}. \tag{5.3}$$

We also call $\pi_2(\mathcal{M}_\eta)$ a *multivariate split* of $\mathcal{M}_\eta$.

We now construct the bipartition model of $\mathcal{S}^*_\eta$. Since similar structured features tend to have similar effects on Y, it is desired to guarantee local contiguity of $\pi_2(\mathcal{S}^*_\eta)$, in the sense that each local cluster in $\mathcal{S}^*_{\eta,j}$ only contains knots that are close to each other with respect to distance d_g.

Spanning tree partition models have recently been proposed as an effective tool to model contiguous partitions of graphs (Li and Sang, 2019; Teixeira et al., 2019; Luo et al., 2021b). They simplify the complicated combinatorial problem of graph partitions by representing partitions as connected components induced by pruning an edge from a spanning tree of the graph. However, there exist some major challenges that prevent us from directly applying these methods to model the bipartition of $\mathcal{S}^*_\eta$. Specifically, the original spanning tree partition models consider a fixed set of vertices at the observed locations. In our case, $\mathcal{S}^*_\eta$ is a subset of reference knots that varies in size and locations as the decision tree node η changes. Moreover, there may exist gaps between local clusters of knots due to interactions between multivariate splits and univariate splits as shown in Figure 5.1(a). If one naively uses an undirected spanning tree graph on the whole reference set $\mathcal{S}^*$ with edges $\mathcal{E}^*$, denoted as $\mathcal{G}^*_T = (\mathcal{S}^*, \mathcal{E}^*)$, removing an arbitrary edge from $\mathcal{G}^*_T$ as in Luo et al. (2021a) does not necessarily lead to a valid $\pi_2(\mathcal{S}^*_\eta)$. Constructing a different spanning tree for each $\mathcal{S}^*_\eta$ is not an ideal alternative either due to the expensive computational costs.

In this chapter, we propose a new bipartition model for $\mathcal{S}^*_\eta$ that is still based on $\mathcal{G}^*_T$ but with a different edge removal rule. Specifically, we consider a spanning tree, $\mathcal{G}^*_T$, where each knot is only connected to its near neighbors with respect to d_g so that it represents the topology of the structured feature space. For instance, $\mathcal{G}^*_T$ can be specified as the minimum spanning tree (MST) of a graph $\mathcal{G}^*$ on $\mathcal{S}^*$ using edge lengths under d_g as edge weights. Following Luo et al. (2021a), $\mathcal{G}^*$ can be constructed using constrained Delaunay triangulations (CDTs; Lee and Schachter, 1980) for constrained domains in $\mathbb{R}^2$ or K nearest neighbor graphs with respect to distance d_g for general manifolds. See Section D.1 for more discussion on constructing $\mathcal{G}^*$ in practice.

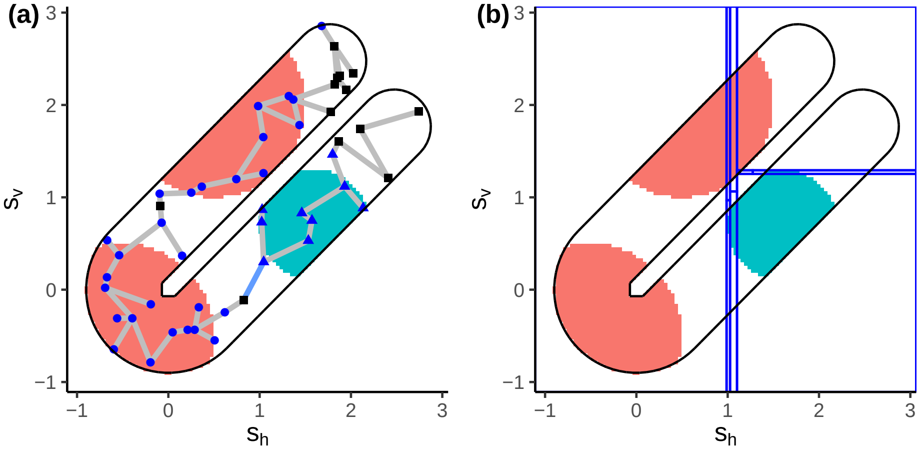

Figure 5.1: (a) A spanning tree graph $\mathcal{G}_T^*$ on reference knots and a colored subset $\mathcal{M}_\eta$ of a U-shaped domain. A bipartition of $\mathcal{S}_\eta^*$ (marked by blue points) after the blue edge is removed from $\mathcal{G}_T^*$ is shown by different point shapes. A multivariate bipartition decision of $\mathcal{M}_\eta$ induced by $\pi_2(\mathcal{S}_\eta^*)$ is marked by different colors. (b) A univariate decision tree partition that approximates $\pi_2(\mathcal{M}_\eta)$, where blue lines represent decision boundaries.

Instead of randomly removing an edge from $\mathcal{E}^*$, we consider a path in $\mathcal{G}_T^*$ connecting two distinct knots in $\mathcal{S}_\eta^*$, which is unique as $\mathcal{G}_T^*$ is a spanning tree. If an edge e^* in the path is removed from $\mathcal{G}_T^*$, we obtain two connected components in the resulting subgraph, which naturally defines a valid bipartition of $\mathcal{S}_\eta^*$ by letting $\mathcal{S}_{\eta,k}^* = \mathcal{S}_\eta^* \cap \mathcal{C}_k^*$, where $\mathcal{C}_k^*$ is the vertices in the kth connected component of $\mathcal{G}_T^*$, and therefore induces a multivariate split of $\mathcal{M}_\eta$. Note that the endpoints of e^* may not belong to $\mathcal{S}_\eta^*$ as $\mathcal{M}_\eta$ can be disconnected. This property motivates a generative prior model for $\pi_2(\mathcal{M}_\eta)$ to be introduced in Section 5.2.2. Figure 5.1(a) illustrates an example of a spanning tree bipartition $\pi_2(\mathcal{S}_\eta^*)$ and the induced $\pi_2(\mathcal{M}_\eta)$, where $\mathcal{M}_\eta$ is a disconnected subset of a U-shape domain. Note that a similar partition in Figure 5.1(b) given by a univariate decision tree has more splits. Note also that the spanning tree bipartition fully respects the intrinsic geometry of $\mathcal{M}$, while a univariate split does not.

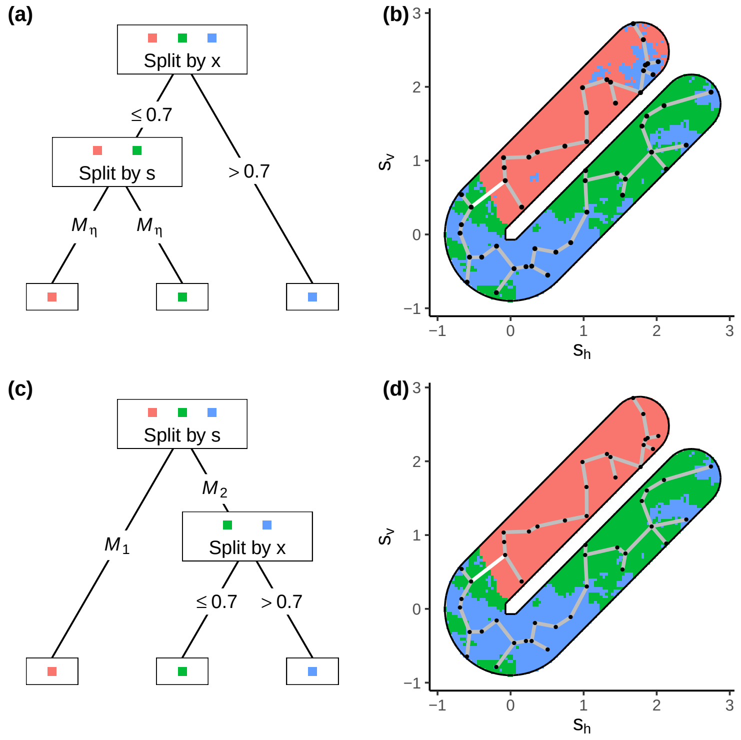

Figure 5.2: (a, c) Two psMDTs with input domain $\mathcal{D} \subseteq \mathcal{M} \times \mathbb{R}$, where $\mathcal{M}$ is a two-dimensional U-shaped domain. (b, d) Partitions of $\mathcal{D}$ projected onto $\mathcal{M}$ corresponding to the psMDTs in (a) and (c). The spanning tree edges removed in multivariate splits are marked in white.

5.2.2 Partially Structured Multivariate Decision Trees

The spanning tree based splits developed in Section 5.2.1 can serve as building blocks for a new class of MDTs involving structured features, called *partially structured MDTs* (psMDTs). A psMDT recursively divides the joint input space $\mathcal{D} \subseteq \mathcal{M} \times \mathcal{X}$ into subsets represented by tree nodes. Note that $\mathcal{D}$ may not equal to the product space of $\mathcal{M}$ and $\mathcal{X}$, because $\mathbf{x}$ and $\mathbf{s}$ may not be independent. Let η be a non-terminal node in a psMDT and η_1 and η_2 be its two offspring nodes. In a psMDT, η either performs a multivariate split using *all* structured features, or a univariate split using one of the unstructured features, to divide the associated subset $\mathcal{D}_\eta \subseteq \mathcal{D}$ into $\pi_2(\mathcal{D}_\eta) = \{\mathcal{D}_{\eta,1}, \mathcal{D}_{\eta,2}\}$ corresponding to η_1 and η_2, respectively. We remark that interaction effects between structured and unstructured features can be naturally captured when the hierarchical splitting of psMDTs involves both

$\mathbf{s}$ and $\mathbf{x}$.

Multivariate splits using structured features. A multivariate split divides $\mathcal{D}_\eta$ by bipartitioning $\mathcal{M}_\eta$, the projection of $\mathcal{D}_\eta$ onto $\mathcal{M}$. For a given $\mathcal{M}_\eta$, we follow the method described in Section 5.2.1 to first split the corresponding $\mathcal{S}_\eta^*$ into $\pi_2(\mathcal{S}_\eta^*)$ to obtain $\pi_2(\mathcal{M}_\eta)$ via (5.2) and (5.3), and then set $\mathcal{D}_{\eta,k} = \mathcal{D}_\eta \cap (\mathcal{M}_{\eta,k} \times \mathcal{X})$ for $k = 1, 2$. Compared to univariate decision trees, the multivariate splits allow psMDTs to generate flexible partitions with a fewer number of nodes. Moreover, since the multivariate splits rely on geodesic distance, psMDTs fully respect the intrinsic geometry and boundary constraints of $\mathcal{M}$.

Univariate splits using unstructured features. In a univariate split, $\mathcal{D}_\eta$ is divided into

$$\mathcal{D}_{\eta,1} = \{(\mathbf{x}, \mathbf{s}) \in \mathcal{D}_\eta : x_{j(\eta)} \le c_\eta\} \tag{5.4}$$

$$\mathcal{D}_{\eta,2} = \mathcal{D}_\eta \setminus \mathcal{D}_{\eta,1}, \tag{5.5}$$

where $x_{j(\eta)}$ is the jth coordinate of $\mathbf{x}$ selected at node η, and c_η is a node-specific cutoff.

Figure 5.2 shows two examples of psMDTs and the partitions they define. Note the hierarchical splits involving both $\mathbf{s}$ and $\mathbf{x}$ may create disconnected $\mathcal{M}_\eta$. Before we introduce the psMDT generating process, we remark that, although both psMDTs, denoted as T, and spanning trees $\mathcal{G}_T^*$ in Section 5.2.1 are referred to as "trees," they are fundamentally different concepts for different purposes. A psMDT is a binary decision tree defining a partition of $\mathcal{D}$, and its vertices/nodes represent subsets of $\mathcal{D}$. On the other hand, $\mathcal{G}_T^*$ encodes an ordering of the multivariate structured knots, and its vertices are the reference knots in $\mathcal{M}$.

Similar to the generative process for univariate decision trees (Chipman et al., 1998), a psMDT can be recursively generated in the following manner:

1. Start with a trivial psMDT that only contains a root node representing the full input space $\mathcal{D}$.

2. Split a terminal node η representing $\mathcal{D}_\eta$ with probability $p_{\text{split}}(\eta)$. If η splits, apply one

of the following split rules to obtain $\pi_2(\mathcal{D}_\eta)$.

(a) With probability p_m, perform a *multivariate* split using the structured features $\mathbf{s}$.

(b) Otherwise, perform a *univariate* split using one of the unstructured features $\mathbf{x}$.

3. Apply Step 2 to each offspring node of η by setting η as η_1 and η_2, respectively.

To generate a multivariate split, we first partition $\mathcal{M}_\eta$ into $\pi_2(\mathcal{M}_\eta)$ by generating a bipartition of $\mathcal{S}_\eta^*$. Motivated by the property in Section 5.2.1, we assume the following generative process of $\pi_2(\mathcal{M}_\eta)$:

1. Randomly sample two distinct knots $\mathbf{s}_i^*$ and $\mathbf{s}_j^*$ from $\mathcal{S}_\eta^*$.

2. Randomly sample an edge e^* from the unique path in $\mathcal{G}_T^*$ connecting $\mathbf{s}_i^*$ and $\mathbf{s}_j^*$.

3. Remove e^* from $\mathcal{G}_T^*$ to obtain $\pi_2(\mathcal{S}_\eta^*)$ and the induced $\pi_2(\mathcal{M}_\eta)$ via (5.2) and (5.3).

Then, we let $\mathcal{D}_{\eta,k} = \mathcal{D}_\eta \cap (\mathcal{M}_{\eta,k} \times \mathcal{X})$ be the subset represented by η's offspring η_k, for $k = 1, 2$.

The generating process of splits using unstructured features follow a similar path as in Chipman et al. (1998) and Denison et al. (1998). Specifically, one of the unstructured features $x_{j(\eta)}$ is randomly chosen, and a random cutoff value c_η is uniformly drawn from its candidate set, which typically depends on the feature and training data. Then we set $\mathcal{D}_{\eta,1}$ and $\mathcal{D}_{\eta,2}$ as in (5.4) and (5.5).

Probability for splits p_{split}. Following Chipman et al. (1998), we specify $p_{\text{split}}(\eta)$ as

$$p_{\text{split}}(\eta) = \alpha(1 + d_\eta)^{-\beta}, \tag{5.6}$$

where d_η is the depth of a node η, and α and β are positive constants. This specification implies that the probability of a node being non-terminal decreases exponentially with its depth and hence implicitly controls the size of a psMDT. We will discuss the choice of α and β in Section 5.3, where we adopt the psMDT generating process as a prior model.

Probability for multivariate splits p_m. This probability controls the portions of multivariate structured splits among all decision tree nodes. The larger p_m is, the more structured information is used for growing a psMDT. When there is no *a priori* information about the true function, $p_m = \max\{d/(d+p), \bar{p}_m\}$ is a reasonable default choice, where $\bar{p}_m$ is chosen to prevent p_m from being dominated by p in high dimensional settings where $p \gg d$.

5.2.3 A Bayesian Sum-of-multivariate-decision-trees Model

A psMDT, T, partitions the input space $\mathcal{D}$ into ℓ disjoint subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_\ell\}$ represented by its ℓ terminal nodes. To apply psMDTs to nonparametric regression tasks, given T and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_\ell)$, we define a piecewise constant mapping from $\mathcal{D}$ to $\mathbb{R}$ as

$$g(\mathbf{s}, \mathbf{x} | T, \boldsymbol{\mu}) = \mu_j, \quad \text{if } (\mathbf{s}, \mathbf{x}) \in \mathcal{D}_j.$$

Using g as a weak learner, a Bayesian additive partially multivariate decision trees (BAMDT) regression model utilizes a summation of piecewise constant functions to approximate the true function f by assuming

$$\mathbb{E}(Y | \mathbf{s}, \mathbf{x}) = \sum_{m=1}^{M} g(\mathbf{s}, \mathbf{x} | T_m, \boldsymbol{\mu}_m),$$

where T_m is a psMDT with ℓ_m terminal nodes, $\boldsymbol{\mu}_m = (\mu_{m1}, \ldots, \mu_{m\ell_m})$ are the terminal node specific constants for T_m, and $M \in \mathbb{N}$ is the pre-specified number of weak learners.

Like other additive tree models such as BART (Chipman et al., 2010) and gradient boosting trees (Friedman, 2001), BAMDT is able to adapt to different smoothness levels and/or discontinuities in the true function. The highly flexible psMDT partitions further allow BAMDT to more effectively capture irregularly shaped decision boundaries where discontinuities or sharp changes happen, while respecting the intrinsic geometry of the structured feature space $\mathcal{M}$.

The regularization prior model of BAMDT is specified in a similar way as BART, which admits the form

$$p\left(\{T_m, \boldsymbol{\mu}_m\}_{m=1}^M, \sigma^2\right) = \left\{\prod_{m=1}^M p(\boldsymbol{\mu}_m|T_m)p(T_m)\right\} p(\sigma^2).$$

The psMDT generating process in Section 5.2.2 is adopted as a prior for T_m's, that is, we assume *a priori* that each T_m is an iid sample from the generating process. We recommend choosing the reference set with a reduced size compared to the number of observations so that computations can be done more efficiently on a reduced spanning tree graph. Nevertheless, this dimension reduction strategy leads to coarser decision tree boundaries in each weak learner. To increase the diversity of psMDTs in the ensemble, we use different sets of reference knots (and hence different spanning trees) for each T_m, which allows each weak learner to explore and learn a different portion of f so that finer discontinuity boundaries in data might be better recovered from ensembles. Following Chipman et al. (2010), we choose $\alpha = 0.95$ and $\beta = 2$ in (5.6), which assigns most of the prior probability to small psMDTs with 2 or 3 nodes and penalizes large T_m's. Shallow psMDTs encourage better mixing and faster convergence in Markov chain Monte Carlo.

Conditional on T_m, we place a conjugate Gaussian prior for $\boldsymbol{\mu}_m$

$$\boldsymbol{\mu}_m|T_m \sim \mathrm{N}_{\ell_m}(\mathbf{0}, \sigma_\mu^2 \mathbf{I}_{\ell_m}),$$

where $\mathbf{I}_\ell$ is an $\ell \times \ell$ identity matrix and $\sigma_\mu^2 = 0.5/(a\sqrt{M})$ with $a > 0$. This prior imposes stronger shrinkage on $\boldsymbol{\mu}_m$ towards zero when we have more weak learners, and therefore prevents overfitting given that we rescale Y into $[-0.5, 0.5]$. We choose $a = 2$ by default, which assigns 0.95 prior probability to $\mathbb{E}(Y|\mathbf{s}, \mathbf{x})$ within $[-0.5, 0.5]$.

The shrinkage prior for $\boldsymbol{\mu}_m$'s, together with the prior for T_m's that favors small psMDTs, ensures that each weak learner only explains a small proportion of response variability, and hence prevent each ensemble membership to be too influential to the overall fit. This therefore regularizes the model to keep it from overfitting the training data.

We complete the prior specification by choosing a conjugate inverse-χ^2 prior for σ^2 in the form of $\sigma^2 \sim \nu\lambda_s/\chi^2_\nu$ for $\lambda_s > 0$ and some degree of freedom ν. We choose $\nu = 3$ and calibrate the prior by selecting λ_s such that $\mathbb{P}(\sigma^2 < \hat{\sigma}^2) = 0.90$ *a priori*, where $\hat{\sigma}^2$ is the sample variance of the responses.

5.3 Bayesian Inference

Bayesian inference of BAMDT is based on a tailored backfitting Markov chain Monte Carlo (MCMC) sampler (Hastie and Tibshirani, 2000), which successively draws $(T_1, \boldsymbol{\mu}_1), \ldots, (T_M, \boldsymbol{\mu}_M)$, and σ^2 from their respective full conditional distributions. To sample from $[T_m, \boldsymbol{\mu}_m | -]$, where $-$ stands for all other parameters and the response data $\mathbf{Y} = (Y_1, \ldots, Y_n)$, we first draw T_m from the collapsed full conditional $p(T_m|-) = \int p(T_m, \boldsymbol{\mu}_m|-) d\boldsymbol{\mu}_m$, and then sample $\boldsymbol{\mu}_m$ from $[\boldsymbol{\mu}_m | T_m, -]$. Thanks to the conjugate priors for $\boldsymbol{\mu}_m$ and σ^2, the distributions $[\boldsymbol{\mu}_m | T_m, -]$ and $[\sigma^2|-]$ admit straightforward closed-form expressions, which are detailed in Appendix D.2.

To sample a new psMDT T^*_m from $p(T_m|-)$, we randomly *grow* or *prune* the existing T_m with equal probability to obtain a tree proposal. In a growing move, one of T_m's terminal nodes, denoted by η, is randomly chosen and split into two offspring nodes following Step 2 of the psMDT generating process in Section 5.2.1. A pruning step does the opposite by first randomly selecting a node with two terminal offspring and then removing its children. The proposed T^*_m is then accepted or rejected following standard Metropolis-Hastings (MH) procedure, and we leave the details to Appendix D.2. Note that the MH acceptance probability involves a likelihood ratio $\mathcal{L}(\mathbf{Y}|T^*_m, -)/\mathcal{L}(\mathbf{Y}|T_m, -)$, where $\mathcal{L}(\mathbf{Y}|T_m, -)$ is the likelihood with $\boldsymbol{\mu}_m$ integrated out. This ratio can be evaluated using its analytical form, thanks to the conjugate Gaussian prior for $\boldsymbol{\mu}_m$. The time complexity to draw T_m is $\mathcal{O}(\max\{n, t\})$ since we utilize a spanning tree that has $t - 1$ edges for multivariate splits.

Using posterior draws of $\{(T_m, \boldsymbol{\mu}_m)\}_{m=1}^M$, we can perform prediction for Y_{new} given $(\mathbf{s}_{\text{new}}, \mathbf{x}_{\text{new}})$. A posterior sample of $\mathbb{E}(Y_{\text{new}}|\mathbf{s}_{\text{new}}, \mathbf{x}_{\text{new}})$ is obtained by summing $g(\mathbf{s}_{\text{new}}, \mathbf{x}_{\text{new}}|T_m, \boldsymbol{\mu}_m)$ over $m = 1, \ldots, M$. A point predictor of Y_{new} can be taken as the posterior mean of $\mathbb{E}(Y_{\text{new}}|\mathbf{s}_{\text{new}}, \mathbf{x}_{\text{new}})$ draws.

Similar to BART, BAMDT offers a natural importance metric for variable selection based on MCMC samples. Let r_z be a split rule involving feature z, where z can be $\mathbf{s}$ or one coordinate of $\mathbf{x} = (x_1, \ldots, x_p)$. For $z = \mathbf{s}$, r_z corresponds to a multivariate split; when $z = x_j$, r_z refers to a univariate split on x_j. The relative importance of z is measured by the proportion of r_z used in the sum-of-psMDT model, denoted by v_z. We use the posterior mean of v_z as a metric to evaluate the importance of z. A higher metric indicates that z is more favored in model fitting, and thus it is more likely that z provides more information for predicting Y.

5.4 Experiments

5.4.1 Simulation Studies

We demonstrate the performance of BAMDT using some synthetic data. The structured feature space $\mathcal{M}$ that we consider is a two-dimensional U-shaped domain as shown in Figure 5.3(a) that is divided into three subsets by a circle centered at the origin with radius 0.9. We generate $n = 500$ uniform random locations in $\mathcal{M}$. The geodesic distance on $\mathcal{M}$ is approximated using the method in Section D.1. The unstructured feature space is set to be $\mathcal{X} \subseteq [0, 1]^p$ with $p \in \{2, 10\}$ (but only one coordinate of $\mathbf{x}$ is involved in the true data generating process). We independently generate x_j for $j = 1, \ldots, p$, but introduce spatial dependence among locations within each x_j to mimic real applications. Using the same data generating scheme, we also simulate features for a test data set of size $n_{\text{test}} = 200$.

As shown in Figure 5.3(a), we consider a true piecewise smooth function defined on $\mathcal{D}$, where we design two jumps across the surfaces $\{(s_h, s_v) \in \mathcal{M} : s_h^2 + s_v^2 = 0.9^2\} \times \mathcal{X}$. The true function only depends on $(\mathbf{s}, x_1)$ and their interaction. The responses in the training and test data sets are generated according to (5.1), where we consider different noise levels $\sigma \in \{0.1, 0.5\}$. We simulate 50 replicates for each level of p and noise. Detailed data generating process can be found in Section D.3.1.

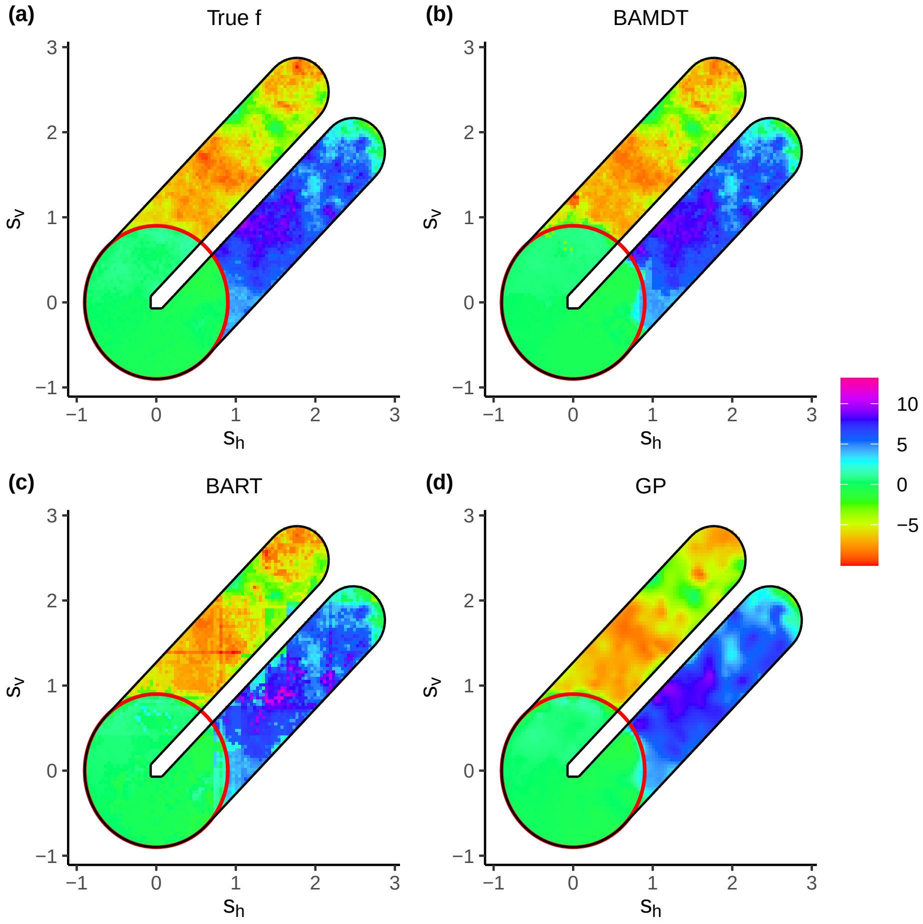

Figure 5.3: (a) True $f(\mathbf{s}, \mathbf{x})$ on a two-dimensional U-shaped domain $\mathcal{M}$ in the setting of $p = 2$. (b-d) Predictive surfaces $\hat{f}(\mathbf{s}, \mathbf{x})$ of BAMDT, BART, and GP regression using one data set with $\sigma = 0.1$. Red circles indicate discontinuity surfaces in the true function projected to $\mathcal{M}$.

We use $M = 50$ weak learners in BAMDT. For each weak learner, we randomly sample $t = 100$ locations from the training data as reference knots. We construct spatial graphs $\mathcal{G}^*$ on reference knots using CDTs following Luo et al. (2021a), and choose their MSTs based on geodesic distance as the spanning trees for multivariate split rules. We use 100 equally spaced grid points as candidates of univariate split cutoffs for each unstructured feature. The probability of performing a multivariate split is set to be $p_m = 2/(2+p)$. We run the MCMC algorithm for $30,000$ iterations, discarding the first half and retaining samples every 10 iterations.

We compare BAMDT with BART and spatial GP regression (see, e.g., Gelfand et al., 2010) that are implemented in `R` packages `BART` (McCulloch et al., 2019) and `GpGp` (Guinness, 2018), respectively, since we focus on the methods that can provide uncertainty measures. The input features of BART include $\mathbf{x}$ and the Cartesian coordinates of $\mathbf{s}$. For BART, we use the same number of weak learners and the same MCMC setting as in BAMDT, and all other hyperparameters are set to be the default values. For the GP regression, the mean function is specified as a linear function of $\mathbf{x}$ and the covariance kernel is chosen to be isotropic Matérn.

We evaluate prediction performance of BAMDT and its competitors using the test data set. Point predictors of BAMDT and BART are based on posterior means, while the one for GP regression is the krigging mean. We use mean square prediction error (MSPE) and mean absolute prediction error (MAPE) to measure point prediction accuracy. We also compare the accuracy of probabilistic prediction using continuous ranked probability scores (CRPSs; Gneiting and Raftery, 2007). For the two Bayesian models BAMDT and BART, CRPS is computed using posterior samples of $\mathbb{E}(Y_{\text{new}}|\mathbf{s}_{\text{new}}, \mathbf{x}_{\text{new}})$; for GP regression, CRPS is evaluated using the kriging distribution. For all the metrics, lower values indicate better performance.

Table 5.1 summarizes the average prediction performance of BAMDT and its competitors over 50 replicates in different settings. In all settings, BAMDT outperforms other methods in terms of all performance metrics. In particular, the comparison between BAMDT and

BART suggests that the proposed MDTs enhance the performance in complex restricted domains while inheriting BART's feature selection capacity. Indeed, the feature importance metric from BAMDT can better identify the truly relevant features $(\mathbf{s}, x_1)$ compared with BART. As an example, in the setting of $p = 10$ and $\sigma = 0.1$, the average percentage of splits involving $(\mathbf{s}, x_1)$ in BAMDT is 73.98%, while the one in BART is 54.62%.

To better examine the prediction from all the models, we present the mean predictive surfaces (as a function of $\mathbf{s}$) in Figure 5.3(b-d) from the models fitted using one randomly selected data set with $p = 2$ and $\sigma = 0.1$. All three models can recover the general pattern of the true function, but the result from BAMDT matches the ground truth best. BAMDT performs fairly well in the interior of each subregion of $\mathcal{M}$, while there are some visible errors around the discontinuity surfaces marked by the red circle, which are expected due to larger uncertainties in data around discontinuities. The predictive surface from BART displays some artificial rectangular decision boundaries such as those in the upper arm, due to the sole use of univariate split rules. There is also some noticeable "leakage" effect in the prediction of BART as evidenced by the underestimation in some regions in the lower arm that are near the upper arm. These undesired patterns are overcome in BAMDT thanks to the use of the multivariate split rules that can generate flexible shaped partition and respect the domain boundary. Unlike BAMDT or BART, the predictive surface from GP regression is too smooth relative to the truth and loses some small scale spatial patterns. We have also compared the predictive uncertainty at different spatial locations in Section D.3.2. Our result suggests that the discontinuity surfaces in the true function are characterized by higher uncertainty from BAMDT, while this pattern does not appear in BART or GP regression. We have also included sensitivity analysis of BAMDT and discussed the results in Section D.3.2.

5.4.2 Application to Sacramento Housing Price Data

We apply BAMDT to analyze housing price data in Sacramento County, California, available in `R` package `caret` (Kuhn, 2021). We focus on $n = 405$ data points from Cities

Table 5.1: Prediction performance of BAMDT and its competing methods in simulations. Standard errors are in parentheses.

			BAMDT	BART	GP
$p = 2$	$\sigma = 0.1$	MSPE	**0.374** (0.015)	1.405 (0.035)	0.620 (0.002)
		MAPE	**0.281** (0.004)	0.612 (0.008)	0.499 (0.001)
		Mean CRPS	**0.219** (0.003)	0.508 (0.007)	0.398 (0.001)
	$\sigma = 0.5$	MSPE	**0.685** (0.011)	1.679 (0.035)	0.949 (0.009)
		MAPE	**0.567** (0.004)	0.829 (0.007)	0.723 (0.004)
		Mean CRPS	**0.438** (0.003)	0.656 (0.006)	0.528 (0.003)
$p = 10$	$\sigma = 0.1$	MSPE	**0.495** (0.022)	1.219 (0.027)	0.662 (0.002)
		MAPE	**0.317** (0.005)	0.688 (0.009)	0.545 (0.001)
		Mean CRPS	**0.252** (0.005)	0.552 (0.008)	0.415 (0.001)
	$\sigma = 0.5$	MSPE	**0.756** (0.018)	1.580 (0.030)	1.008 (0.010)
		MAPE	**0.584** (0.005)	0.878 (0.008)	0.754 (0.004)
		Mean CRPS	**0.453** (0.005)	0.686 (0.007)	0.544 (0.003)

of Sacramento and Elk Grove. The observed housing price and city boundary* are shown in Figure 5.4(a). Note that the City of Sacramento is divided by the American River near 38.6°N. We model the logarithm of housing price (in U.S. dollars) as a function of the house location (in latitude and longitude), number of bedrooms, number of bathrooms, and square footage. We treat the location as a structured feature $\mathbf{s}$ and all other covariates as unstructured features $\mathbf{x}$. The goal is to examine BAMDT's performance in predicting housing prices with new features.

We fit BAMDT, BART, and spatial GP regression to the data. The settings of them are identical to those in Section 5.4.1, except that we use $t = 150$ knots for BAMDT.

We first compare prediction performance of the three models using 5-fold cross-validation. Table 5.2 shows the performance metrics computed using the original price scale (instead of log scale). BAMDT achieves better prediction accuracy than the other two methods in all metrics.

Next, we turn to the mean predictive surface from each model fitted using all the observations. We consider a representative house with median unstructured features, namely,

*City shape file is retrieved from Sacramento County GIS (2015)

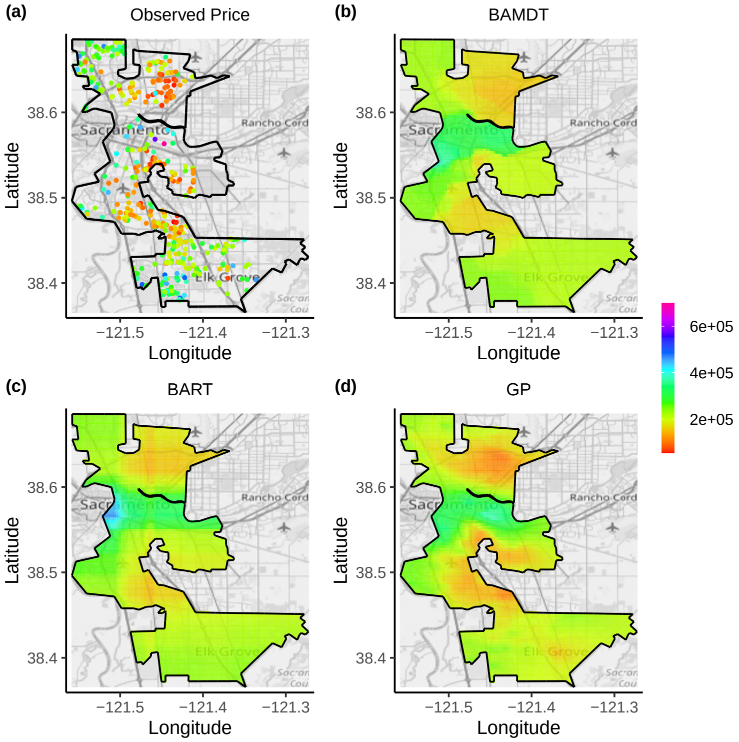

Figure 5.4: (a) Observed housing price (in U.S. dollars). (b-d) Predicted price for a representative house from BAMDT, BART, and GP regression.

three bedrooms, two bathrooms, and 1436 square feet. We display its predicted price at different locations in Figure 5.4(b-d) to examine the marginal spatial effect on housing prices. The predictive surfaces from BART and GP regression fail to respect the boundary constraints, especially near the American River. In contrast, there is a clear jump across the river in the surface from BAMDT. As in the simulation studies, BART only identifies axis-parallel discontinuities, while BAMDT could detect more flexible discontinuity boundaries with meaningful interpretations such as the one along U.S. Highway 50. Compared with BAMDT and BART, the GP regression tends to give lower predictions in the regions of low housing price, possibly due to the lack of interaction between $\mathbf{s}$ and $\mathbf{x}$ in the model, and its predicted price changes smoothly near U.S. Highway 50. We have also examined feature importance and the marginal effect of square footage in Appendix D.4.

Finally, we examine the prediction uncertainty for the representative house. Figure 5.5

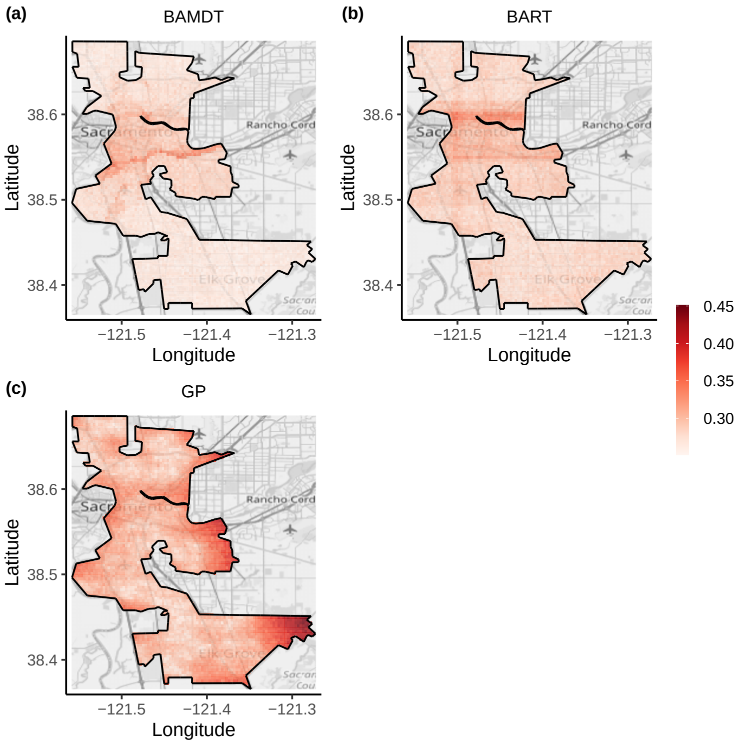

Figure 5.5: Posterior predictive standard deviation of log-price for a representative house from (a) BAMDT, (b) BART, and (c) GP regression.

Table 5.2: Prediction performance of BAMDT and its competing methods in Sacramento housing data set.

	BAMDT	BART	GP
Root MSPE	**62128**	64607	69701
MAPE	**43110**	45224	48790
Mean CRPS	**34107**	35940	35633

shows the predictive standard deviation of *log-price* from the three models. There is a narrow band with high uncertainty near U.S. Highway 50 and Sacramento Zoo from BAMDT that separates the downtown area and East Sacramento from southern regions. This band corresponds to an abrupt price change in Figure 5.4(b), and thus it is associated with higher prediction uncertainty.

5.5 Conclusion and Discussion

In this chapter, we proposed a new Bayesian additive decision tree model for structured regressions. The method relaxes the limitations of conventional BART methods due to axis-parallel split rules by allowing a flexible mixture of univariate and multivariate split rules in decision tree weak learners. The proposed multivariate split rules are built upon a manifold bipartition model via predictive spanning trees that is capable of complying to intrinsic geometry and boundary constraints of the structured feature space.

Thanks to its Bayesian nature, BAMDT is promising to serve as a flexible nonparametric prior for modeling latent functions in various hierarchical modeling settings. The method has great potential beyond predictive regression tasks to other machine learning tasks such as classification, density estimation, survival analysis, and causal inference, to name a few. Besides these extensions, future research may also include theoretical investigations of function approximation performance via Bayesian posterior concentration theories, and computational accelerations via extensions of informed MCMC and importance sampling (Zanella and Roberts, 2019; Griffin et al., 2021), spike and slab lasso (Ročková and George, 2018), or variational Bayes inference (Blei et al., 2017).

6. CONCLUSION

Motivated by many applications involving complex spatial data, this dissertation has studied Bayesian spanning-tree-based models from four aspects. First, we introduce a *probabilistic* spanning tree partition model on a finite number of spatial locations to address the problem of learning spatially clustered relationship between the response variable and the covariates. Then we extend the finite partition model to the spatial domain by developing a *soft* stochastic partition process, which serves as a building block for a new class of locally stationary Gaussian process (GP) models. Third, we consider nonparametric regression with structured features (e.g., spatial locations) on a complex constrained domain, and propose a Bayesian additive ensemble model based on random spanning tree (RST) manifold partitions. Finally, a novel partially structured multivariate decision tree model is developed for nonparametric ensemble learning by incorporating both predictive-spanning-tree-based multivariate decision rules and univariate decision rules. The main results of this dissertation are summarized as follows.

In Chapter 2, we propose an RST partition model on a finite set of spatial locations that enjoys highly flexible cluster shapes and sizes and guarantees spatial contiguity of the clusters. Utilizing it as a prior model, a Bayesian spatially clustered coefficient (BSCC) regression model is developed with an efficient MCMC sampler. We study the theoretical properties of the RST partition model and derive posterior concentration results of BSCC. The superior performance of BSCC is demonstrated via simulation studies and the temperature-salinity data from the Atlantic Ocean.

In Chapter 3, a soft partition process is introduced to generalize the partition model on a *fixed* finite location set to a stochastic process, based upon which we propose a legitimate locally stationary GP model. Combined with a predictive RST space partition prior, we develop a spanning-treed GP regression model and establish its posterior consistency and contraction results. The performance of the proposed model is examined using simulated data and the precipitation data in the contiguous United States.

Chapter 4 and Chapter 5 study nonparametric regression problems on non-trivial spatial domains. In Chapter 4, we focus on nonparametric regression with structured features only. We extend the flexible RST partition model to a finite set of spatial locations on a compact Riemannian manifold $\mathcal{M}$ such that the intrinsic geometries and domain boundary constraints of $\mathcal{M}$ are fully respected. Using this partition model as a weak learner, a new Bayesian ensemble model, called the Bayesian additive regression spanning trees (BAST), is proposed for nonparametric function estimation and prediction on complex constrained domains. We apply BAST to constrained domains in $\mathbb{R}^2$ and manifolds embedded in $\mathbb{R}^3$ to illustrate its utility.

In Chapter 5, we address a more general nonparametric regression problem involving *both* structured and unstructured features, where the structured features possibly lie on a compact Riemannian manifold. We introduce a novel multivariate split rule using structured features based on predictive spanning tree manifold bipartitions that can fully respect the intrinsic geometries of the structured feature space. We develop a partially structured multivariate decision tree (psMDT) that uses multivariate split rules for structured features and univariate split rules for possibly high dimensional unstructured features. A Bayesian additive psMDT model is then proposed and demonstrated to achieve good performance in synthetic data and a real housing price data set.

REFERENCES

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679.

Armagan, A., Dunson, D. B., Lee, J., Bajwa, W. U., and Strawn, N. (2013). Posterior consistency in linear models under shrinkage priors. *Biometrika*, 100(4):1011–1018.

Assunção, R. M., Neves, M. C., Câmara, G., and da Costa Freitas, C. (2006). Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20(7):797–811.

Aswani, A., Bickel, P., and Tomlin, C. (2011). Regression on manifolds: Estimation of the exterior derivative. *The Annals of Statistics*, 39(1):48–81.

Aydin, O., Janikas, M. V., Assunção, R., and Lee, T.-H. (2018). SKATER-CON: Unsupervised regionalization via stochastic tree partitioning within a consensus framework using random spanning trees. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 33–42.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014). *Hierarchical Modeling and Analysis for Spatial Data.* Chapman and Hall/CRC.

Banerjee, S., Gelfand, A., Knight, J. R., and Sirmans, C. (2004). Spatial modeling of house prices using normalized distance-weighted sums of stationary processes. *Journal of Business & Economic Statistics*, 22(2):206–213.

Barron, A. R. (1998). Information-theoretic characterization of bayes performance and the choice of priors in parametric and nonparametric problems. In Bernardo, J., Burger, J., and Smith, A., editors, *Bayesian Statistics 6*, pages 27–52. Oxford University Press.

Bhattacharya, A., Pati, D., and Dunson, D. (2014). Anisotropic function estimation using multi-bandwidth Gaussian processes. *Annals of statistics*, 42(1):352.

Blaser, R. and Fryzlewicz, P. (2016). Random rotation ensembles. *The Journal of Machine Learning Research*, 17(1):126–151.

Blei, D. M. and Frazier, P. I. (2011). Distance dependent Chinese restaurant processes. *Journal of Machine Learning Research*, 12(Aug):2461–2488.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877.

Bolin, D., Wallin, J., and Lindgren, F. (2019). Latent Gaussian random field mixture models. *Computational Statistics & Data Analysis*, 130:80–93.

Bonato, V., Baladandayuthapani, V., Broom, B. M., Sulman, E. P., Aldape, K. D., and Do, K.-A. (2011). Bayesian ensemble methods for survival prediction in gene expression data. *Bioinformatics*, 27(3):359–367.

Borovitskiy, V., Azangulov, I., Terenin, A., Mostowsky, P., Deisenroth, M. P., and Durrande, N. (2021). Matérn Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. P. (2020). Matérn Gaussian processes on Riemannian manifolds. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.

Cañete-Sifuentes, L., Monroy, R., and Medina-Pérez, M. A. (2021). A review and experimental comparison of multivariate decision trees. *IEEE Access*.

Castro, R., Willett, R., and Nowak, R. (2005). Faster rates in regression via active learning. In *NIPS*, volume 18, pages 179–186.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794.

Chen, Y., Davis, T. A., Hager, W. W., and Rajamanickam, S. (2008). Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM*

Transactions on Mathematical Software (TOMS), 35(3):22.

Cheng, M.-Y. and Wu, H.-T. (2013). Local linear regression on manifolds and its geometric interpretation. *Journal of the American Statistical Association*, 108(504):1421–1434.

Chew, L. P. (1989). Constrained Delaunay triangulations. *Algorithmica*, 4(1):97–108.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Chu, T., Zhu, J., and Wang, H. (2011). Penalized maximum likelihood estimation and variable selection in geostatistics. *The Annals of Statistics*, 39(5):2607–2625.

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812.

Denison, D. G. and Holmes, C. C. (2001). Bayesian partitioning for estimating disease risk. *Biometrics*, 57(1):143–149.

Denison, D. G., Mallick, B. K., and Smith, A. F. (1998). A Bayesian CART algorithm. *Biometrika*, 85(2):363–377.

Deshpande, S. K., Bai, R., Balocchi, C., Starling, J. E., and Weiss, J. (2020). VCBART: Bayesian trees for varying coefficients. *arXiv preprint arXiv:2003.06416*.

Di Marzio, M., Panzera, A., and Taylor, C. C. (2014). Nonparametric regression for spherical data. *Journal of the American Statistical Association*, 109(506):748–763.

Diestel, R. (2016). *Graph Theory*. Electronic library of mathematics. Springer, 5th edition.

Dunson, D. B., Wu, H.-T., and Wu, N. (2020). Diffusion based Gaussian processes on restricted domains. *arXiv preprint arXiv:2010.07242*.

Fan, X., Li, B., Luo, L., and Sisson, S. A. (2021). Bayesian nonparametric space partitions: A survey. In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4408–4415. International Joint Conferences on

Artificial Intelligence Organization. Survey Track.

Feng, W., Lim, C. Y., Maiti, T., and Zhang, Z. (2016). Spatial regression and estimation of disease risks: A clustering-based approach. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 9(6):417–434.

Fotheringham, S., Brunsdon, C., and Charlton, M. (2003). *Geographically Weighted Regression.* John Wiley & Sons, Chichester.

Fouedjio, F. (2017). Second-order non-stationary modeling approaches for univariate geostatistical data. *Stochastic environmental research and risk assessment*, 31(8):1887–1906.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Gangnon, R. E. and Clayton, M. K. (2000). Bayesian detection and modeling of spatial disease clustering. *Biometrics*, 56(3):922–935.

Gao, L., Datta, A., and Banerjee, S. (2021). Hierarchical multivariate directed acyclic graph auto-regressive (mdagar) models for spatial diseases mapping. *arXiv preprint arXiv:2102.02911*.

Gelfand, A. E., Diggle, P., Guttorp, P., and Fuentes, M. (2010). *Handbook of spatial statistics*. CRC press.

Gelfand, A. E., Kim, H.-J., Sirmans, C., and Banerjee, S. (2003). Spatial modeling with spatially varying coefficient processes. *Journal of the American Statistical Association*, 98(462):387–396.

Gelfand, A. E., Kottas, A., and MacEachern, S. N. (2005). Bayesian nonparametric spatial modeling with Dirichlet process mixing. *Journal of the American Statistical Association*, 100(471):1021–1035.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and computing*, 24(6):997–1016.

Gerber, F. and Nychka, D. W. (2021). Parallel cross-validation: A scalable fitting method for Gaussian process models. *Computational Statistics & Data Analysis*, 155:107113.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163. Interface Foundation of North America.

Ghosal, S., Ghosh, J. K., and van der Vaart, A. W. (2000). Convergence rates of posterior distributions. *Annals of Statistics*, pages 500–531.

Ghosal, S. and Roy, A. (2006). Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 34(5):2413–2429.

Ghosh, J. K. and Ramamoorthi, R. (2003). *Bayesian nonparametrics.* Springer Science & Business Media.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378.

Gosoniu, L. and Vounatsou, P. (2011). Non-stationary partition modeling of geostatistical data for malaria risk mapping. *Journal of Applied Statistics*, 38(1):3–13.

Gramacy, R. B. (2007). tgp: an R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models. *Journal of Statistical Software*, 19(9):6.

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.

Gramacy, R. B. and Taddy, M. (2010). Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models. *Journal of Statistical Software*, 33(6):1–48.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Green, P. J. and Sibson, R. (1978). Computing Dirichlet tessellations in the plane. *The Computer Journal*, 21(2):168–173.

Griffin, J., Latuszyński, K., and Steel, M. (2021). In search of lost mixing time: adaptive

Markov chain Monte Carlo schemes for Bayesian variable selection with very large p. *Biometrika*, 108:53–69.

Grygorash, O., Zhou, Y., and Jorgensen, Z. (2006). Minimum spanning tree based clustering algorithms. In *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pages 73–81. IEEE.

Guan, Y. and Stephens, M. (2011). Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *The Annals of Applied Statistics*, pages 1780–1815.

Guella, J. C., Menegatto, V. A., and Porcu, E. (2018). Strictly positive definite multivariate covariance functions on spheres. *Journal of Multivariate Analysis*, 166:150–159.

Guinness, J. (2018). Permutation and grouping methods for sharpening Gaussian process approximations. *Technometrics*, 60(4):415–429.

Guinness, J. and Fuentes, M. (2016). Isotropic covariance functions on spheres: Some properties and modeling considerations. *Journal of Multivariate Analysis*, 143:143–152.

Guo, D. (2008). Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). *International Journal of Geographical Information Science*, 22(7):801–823.

Hahn, P. R., Murray, J. S., and Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, 15(3):965–1056.

Hastie, T. and Tibshirani, R. (2000). Bayesian backfitting (with comments and a rejoinder by the authors). *Statistical Science*, 15(3):196–223.

He, J., Yalov, S., and Hahn, P. R. (2019). XBART: Accelerated Bayesian additive regression trees. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1130–1138. PMLR.

Heaton, M. J., Christensen, W. F., and Terres, M. A. (2017). Nonstationary Gaussian process models using spatial hierarchical clustering from finite differences. *Technometrics*, 59(1):93–101.

Hegarty, A. and Barry, D. (2008). Bayesian disease mapping using product partition models. *Statistics in Medicine*, 27(19):3868–3893.

Henry, G. and Rodriguez, D. (2009). Robust nonparametric regression on Riemannian manifolds. *Journal of Nonparametric Statistics*, 21(5):611–628.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Jeong, J. and Jun, M. (2015). A class of Matérn-like covariance functions for smooth processes on a sphere. *Spatial Statistics*, 11:1–18.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154.

Kim, H.-M., Mallick, B. K., and Holmes, C. C. (2005). Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association*, 100(470):653–668.

Knorr-Held, L. and Raßer, G. (2000). Bayesian detection of clusters and discontinuities in disease maps. *Biometrics*, 56(1):13–21.

Konomi, B. A., Sang, H., and Mallick, B. K. (2014). Adaptive Bayesian nonstationary modeling for large spatial datasets using covariance approximations. *Journal of Computational and Graphical Statistics*, 23(3):802–829.

Kowal, D. R., Matteson, D. S., and Ruppert, D. (2019). Dynamic shrinkage processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(4):781–804.

Kpotufe, S. (2011). k-NN regression adapts to local intrinsic dimension. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Kuhn, M. (2021). *caret: Classification and Regression Training.* R package version 6.0-90.

Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics-Theory and Methods*, 26(6):1481–1496.

Kulldorff, M. and Nagarwalla, N. (1995). Spatial disease clusters: detection and inference. *Statistics in Medicine*, 14(8):799–810.

Lai, M.-J. and Schumaker, L. L. (2007). *Spline functions on triangulations*, volume 110. Cambridge University Press.

Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338.

Lee, D.-T. and Schachter, B. J. (1980). Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242.

Lee, J., Gangnon, R. E., and Zhu, J. (2017). Cluster detection of spatial regression coefficients. *Statistics in Medicine*, 36(7):1118–1133.

Li, C. (2020). Bayesian fixed-domain asymptotics: Bernstein-von Mises theorem for covariance parameters in a Gaussian process model. *arXiv preprint arXiv:2010.02126*.

Li, F. and Sang, H. (2019). Spatial homogeneity pursuit of regression coefficients for large datasets. *Journal of the American Statistical Association*, 114(527):1050–1062.

Lin, L., Mu, N., Cheung, P., and Dunson, D. (2019). Extrinsic Gaussian processes for regression and classification on manifolds. *Bayesian Analysis*, 14(3):887–906.

Lin, P.-S. (2014). Generalized scan statistics for disease surveillance. *Scandinavian Journal of Statistics*, 41(3):791–808.

Lin, P.-S., Kung, Y.-H., and Clayton, M. (2016). Spatial scan statistics for detection of multiple clusters with arbitrary shapes. *Biometrics*, 72(4):1226–1234.

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 (Statistical Methodology)*, 73(4):423–498.

Linero, A. R. and Yang, Y. (2018). Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodol-*

ogy), 80(5):1087–1110.

Luo, Z. T., Sang, H., and Mallick, B. (2021a). BAST: Bayesian Additive Regression Spanning Trees for Complex Constrained Domain. *Advances in Neural Information Processing Systems*, 34.

Luo, Z. T., Sang, H., and Mallick, B. (2021b). A Bayesian contiguous partitioning method for learning clustered latent variables. *Journal of Machine Learning Research*, 22(37):1–52.

Ma, Z., Xue, Y., and Hu, G. (2020). Heterogeneous regression models for clusters of spatial dependent data. *Spatial Economic Analysis*, 15(4):459–475.

Madrid Padilla, O. H., Sharpnack, J., Chen, Y., and Witten, D. M. (2020). Adaptive nonparametric regression with the k-nearest neighbour fused lasso. *Biometrika*, 107(2):293–310.

McCulloch, R., Sparapani, R., Gramacy, R., Spanbauer, C., and Pratola, M. (2019). *BART: Bayesian Additive Regression Trees*. R package version 2.7.

Menafoglio, A., Gaetani, G., and Secchi, P. (2018). Random domain decompositions for object-oriented kriging over complex domains. *Stochastic environmental research and risk assessment*, 32(12):3421–3437.

Meng, D., Leung, Y., Xu, Z., Fung, T., and Zhang, Q. (2008). Improving geodesic distance estimation based on locally linear assumption. *Pattern Recognition Letters*, 29(7):862–870.

Mu, J., Wang, G., and Wang, L. (2018). Estimation and inference in spatially varying coefficient models. *Environmetrics*, 29(1):e2485.

Murray, J. S. (2020). Log-linear Bayesian additive regression trees for multinomial logistic and count regression models. *Journal of the American Statistical Association*, (just-accepted):1–35.

Niu, M., Cheung, P., Lin, L., Dai, Z., Lawrence, N., and Dunson, D. (2019). Intrinsic Gaussian processes on complex constrained domains. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(3):603–627.

Osborne, M. A. (2010). *Bayesian Gaussian processes for sequential prediction, optimisation* *and quadrature*. PhD thesis, Oxford University, UK.

Paciorek, C. J. and Schervish, M. J. (2006). Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics: The official journal of the International Environmetrics Society*, 17(5):483–506.

Page, G. L. and Quintana, F. A. (2016). Spatial product partition models. *Bayesian Analysis*, 11(1):265–298.

Park, C., Huang, J., and Ding, Y. (2011). Domain decomposition approach for fast Gaussian process regression of large spatial data sets. *Journal of Machine Learning Research*, 12:1697–1728.

Payne, R. D., Guha, N., Ding, Y., and Mallick, B. K. (2020). A conditional density estimation partition model using logistic Gaussian processes. *Biometrika*, 107(1):173–190.

Pelletier, B. (2006). Non-parametric regression estimation on closed Riemannian manifolds. *Journal of Nonparametric Statistics*, 18(1):57–67.

Penrose, M. D. (1999). A strong law for the longest edge of the minimal spanning tree. *The Annals of Probability*, 27(1):246–260.

Penrose, M. D. (2007). Laws of large numbers in stochastic geometry with statistical applications. *Bernoulli*, 13(4):1124–1150.

Peruzzi, M., Banerjee, S., and Finley, A. O. (2020). Highly scalable Bayesian geostatistical modeling via meshed Gaussian processes on partitioned domains. *Journal of the American Statistical Association*, pages 1–14.

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349.

Pope, C. A., Gosling, J. P., Barber, S., Johnson, J. S., Yamaguchi, T., Feingold, G., and Blackwell, P. G. (2021). Gaussian process modeling of heterogeneity and discontinuities using voronoi tessellations. *Technometrics*, 63(1):53–63.

Preparata, F. P. and Shamos, M. I. (2012). *Computational Geometry: An Introduction.*

Springer Science & Business Media.

Ramsay, T. (2002). Spline smoothing over difficult regions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):307–319.

Risser, M. D. (2016). Nonstationary spatial modeling, with emphasis on process convolution and covariate-driven approaches. *arXiv preprint arXiv:1610.02447*.

Risser, M. D., Calder, C. A., Berrocal, V. J., and Berrett, C. (2019). Nonstationary spatial prediction of soil organic carbon: implications for stock assessment decision making. *The Annals of Applied Statistics*, 13(1):165–188.

Risser, M. D. and Turek, D. (2020). Bayesian inference for high-dimensional nonstationary Gaussian processes. *Journal of Statistical Computation and Simulation*, 90(16):2902–2928.

Ročková, V. and George, E. I. (2018). The spike-and-slab LASSO. *Journal of the American Statistical Association*, 113:431–444.

Ročková, V. and Saha, E. (2019). On theory for BART. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2839–2848. PMLR.

Ročková, V. and van der Pas, S. (2020). Posterior concentration for Bayesian regression trees and forests. *Annals of Statistics*, 48(4):2108–2131.

Sacramento County GIS (2015). City boundaries: Sacramento County, California, 2015. [Shapefile]. Sacramento County GIS. Retrieved from `https://earthworks.stanford.edu/catalog/stanford-kq595nj1377`. Accessed on December 29, 2021.

Sangalli, L. M., Ramsay, J. O., and Ramsay, T. O. (2013). Spatial spline regression models. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 681–703.

Scott-Hayward, L. A. S., MacKenzie, M. L., Donovan, C. R., Walker, C., and Ashe, E. (2014). Complex region spatial smoother (CReSS). *Journal of Computational and Graphical Statistics*, 23(2):340–360.

Shen, Y., Ng, A., and Seeger, M. (2006). Fast gaussian process regression using kd-trees. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, number CONF.

Simmonds, J. G. (2012). *A Brief on Tensor Analysis*. Springer Science &Simmonds Business Media.

Sivaganesan, S., Müller, P., and Huang, B. (2017). Subgroup finding via Bayesian additive regression trees. *Statistics in medicine*, 36(15):2391–2403.

Song, Q. and Cheng, G. (2020). Bayesian fusion estimation via t shrinkage. *Sankhya A*, 82(2):353–385.

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media.

Talley, L. D. (2011). *Descriptive Physical Oceanography: An Introduction*. Academic Press, London.

Tan, Y. V. and Roy, J. (2019). Bayesian additive regression trees and the general BART model. *Statistics in medicine*, 38(25):5048–5069.

Teixeira, L. V., Assunção, R. M., and Loschi, R. H. (2015). A generative spatial clustering model for random data through spanning trees. In *2015 IEEE International Conference on Data Mining*, pages 997–1002. IEEE.

Teixeira, L. V., Assunção, R. M., and Loschi, R. H. (2019). Bayesian space-time partitioning by sampling and pruning spanning trees. *Journal of Machine Learning Research*, 20(85):1–35.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

Vallis, G. K. (2017). *Atmospheric and Oceanic Fluid Dynamics*. Cambridge University Press.

van der Vaart, A. and van Zanten, H. (2011). Information rates of nonparametric Gaussian process methods. *Journal of Machine Learning Research*, 12(6).

van der Vaart, A. W. and van Zanten, J. H. (2008). Rates of contraction of posterior distributions based on Gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463.

Wang, H. and Ranalli, M. G. (2007). Low-rank smoothing splines on complicated domains. *Biometrics*, 63(1):209–217.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(12).

Willett, R., Nowak, R., and Castro, R. M. (2006). Faster rates in regression via active learning. In *Advances in Neural Information Processing Systems*, pages 179–186.

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

Wood, S. N. (2006). *Generalized Additive Models: An Introduction with R.* Chapman & Hall/CRC, Boca Raton, Florida, first edition. ISBN 1-58488-474-6.

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(5):931–955.

Wu, Y., Tjelmeland, H., and West, M. (2007). Bayesian CART: Prior specification and posterior simulation. *Journal of Computational and Graphical Statistics*, 16(1):44–66.

Yang, Y., Wainwright, M. J., and Jordan, M. (2016). On the computational complexity of high-dimensional Bayesian variable selection. *The Annals of Statistics*, 44(6):2497–2532.

Yıldız, O. T. (2011). Model selection in omnivariate decision trees using structural risk minimization. *Information Sciences*, 181(23):5214–5226.

Zahn, C. T. (1970). Graph theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20(SLAC-PUB-0672-REV):68.

Zanella, G. and Roberts, G. (2019). Scalable importance tempering and Bayesian variable selection. *Journal of the Royal Statistical Society Series B*, 81:489–517.

Zhang, B., Sang, H., and Huang, J. Z. (2019). Smoothed full-scale approximation of Gaussian process models for computation of large spatial data sets. *Statistica Sinica*, 29(4):1711–1737.

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465):250–261.

Zhang, L., Guindani, M., Versace, F., and Vannucci, M. (2014). A spatio-temporal nonparametric Bayesian variable selection model of fMRI data for clustering correlated time courses. *NeuroImage*, 95:162–175.

Zhou, Q. and Guan, Y. (2019). Fast model-fitting of Bayesian variable selection regression using the iterative complex factorization algorithm. *Bayesian Analysis*, 14(2):573.

APPENDIX A

SUPPLEMENTARY MATERIALS FOR CHAPTER 2 [*]

A.1 Proofs of Main Results

A.1.1 Proof of Proposition 2.2

To prove Proposition 2.2, we first introduce a lemma.

Lemma A.1. *(Proposition 8.1.1 of Diestel 2016) Every connected graph contains at least one spanning tree.*

Now we prove Proposition 2.2.

Proof of Proposition 2.2. We first construct a subgraph of $\mathcal{G}_0$ and then show that it is a spanning tree that induces π. Consider the following procedure with initial values $t = 1$ and $\mathcal{T}^0 = (\mathcal{T}^0, \mathcal{E}^0) = (\emptyset, \emptyset)$:

1. If $t = 1$, pick an arbitrary vertex $v \in \mathcal{V}_0$; otherwise, pick a vertex $v \in \mathcal{V}_0 \backslash \mathcal{V}^{t-1}$ that is connected to a vertex in $\mathcal{T}^{t-1}$ by an edge e (the existence of v is guaranteed since $\mathcal{G}_0$ is connected). Without loss of generality suppose v belongs to $\mathcal{C}_t$.

2. By Lemma A.1 we know there is a spanning tree $\mathcal{T}^* = (\mathcal{V}^*, \mathcal{E}^*)$ of the subgraph $(\mathcal{C}_t, \mathcal{E}_{\mathcal{C}_t})$, where $\mathcal{E}_{\mathcal{C}_t} \subseteq \mathcal{E}_0$ is the set of edges whose endpoints belong to $\mathcal{C}_t$. If $t = 1$ let $\mathcal{T}^t = \mathcal{T}^*$; otherwise, let $\mathcal{T}^t = (\mathcal{V}^{t-1} \cup \mathcal{V}^*, \mathcal{E}^{t-1} \cup \mathcal{E}^* \cup \{e\})$, where $\mathcal{V}^{t-1}$ and $\mathcal{E}^{t-1}$ are the vertex set and edge set of $\mathcal{T}^{t-1}$, respectively.

3. If $\mathcal{T}^t$ contains all vertices in $\mathcal{G}_0$, then stop; otherwise, let $t := t + 1$ and go to step 1.

[*] Reprinted from Luo, Z. T., Sang, H., and Mallick, B. (2021b). A Bayesian contiguous partitioning method for learning clustered latent variables. *Journal of Machine Learning Research*, 22(37):1–52. The authors hold the copyright.

We show that each $\mathcal{T}^t, t \geq 1$ is a tree by induction arguments. By construction $\mathcal{T}^1$ is a tree. Suppose $\mathcal{T}^{t-1}$ is a tree, then $\mathcal{T}^t$ is also a tree since both $\mathcal{T}^{t-1}$ and $\mathcal{T}^*$ are trees.

Therefore, the final $\mathcal{T}^t$ that contains all vertices of $\mathcal{G}_0$ is a spanning tree and the collection of e's in each iteration is $\mathcal{E}_{k-1}$. This completes the proof of Proposition 2.2. □

A.1.2 Proof of Theorem 2.3

To prove Theorem 2.3 we need some lemmas.

Lemma A.2. *(Lemma 1 of Laurent and Massart 2000) Let χ_d^2 be a chi-square distribution with degree of freedom d. Then the following concentration inequalities hold for any $x > 0$:*

$$\mathbb{P}\left(\chi_d^2 > d + 2x + 2\sqrt{dx}\right) \leq \exp(-x)$$

and

$$\mathbb{P}\left(\chi_d^2 < d - 2\sqrt{dx}\right) \leq \exp(-x).$$

Lemma A.3. *(Lemma 6 of Barron 1998) Let f_θ be the likelihood function with parameter $\theta \in \Theta_n$, $f^* \equiv f_{\theta^*}$ be the true probability density of data generation with true data generation parameter θ^*, $\mathbb{E}_\theta, \mathbb{E}^*$ denote the expectations under θ and θ^* respectively, $\mathbb{P}^*$ denote the probability measure for data generation under θ^*, and Π, Π_n denote the prior distribution on Θ_n with density $\pi(\theta)$ and the posterior, respectively. Let B_n and C_n be two subsets of the parameter space Θ_n, and ϕ_n be a test function satisfying $\phi_n(D_n) \in \{0,1\}$ for any data D_n. If $\Pi(B_n) \leq b_n, \mathbb{E}^*\{\phi(D_n)\} \leq b_n', \sup_{\theta \in C_n} \mathbb{E}_\theta\{1 - \phi(D_n)\} \leq c_n$, and*

$$\mathbb{P}^*\left(\frac{m(D_n)}{f^*(D_n)} \geq a_n\right) \geq 1 - a_n'$$

where $m(D_n) = \int_{\Theta_n} \pi(\theta) f_\theta(D_n) d\theta$ is the marginal likelihood of D_n. Then for any $\Delta_n > 0$,

$$\mathbb{P}^*\left(\Pi_n(C_n \cup B_n \mid D_n) \geq \frac{b_n + c_n}{a_n \Delta_n}\right) \leq \Delta_n + a_n' + b_n'.$$

Next we give the proof of Theorem 2.3. With some abuse of notations we use $i \in \mathcal{C}_j$ to denote that the ith location belongs to the jth cluster and (i, j) to denote the edge connecting $\mathbf{s}_i$ and $\mathbf{s}_j$ throughout the proof. We also denote the L_1 and supremum norm by $\|\cdot\|_1$ and $\|\cdot\|_\infty$, respectively.

Proof of Theorem 2.3. Given an arbitrary partition π with k clusters, for the jth cluster, we define an estimator as

$$\hat{\beta}_{(j)} = \frac{\sum_{i \in \mathcal{C}_j} x_i y_i}{\sum_{i \in \mathcal{C}_j} x_i^2},$$

where $y_i = y(\mathbf{s}_i)$. Further define $\hat{\beta}_\pi(\mathbf{y}) \in \mathbb{R}^n$ such that the ith element $\hat{\beta}_{\pi,i}(\mathbf{y}) = \hat{\beta}_{(j)}$ if $i \in \mathcal{C}_j$ under π, and $\hat{\sigma}^2_\pi(\mathbf{y}) = \|\mathbf{y} - \hat{\boldsymbol{\mu}}_\pi(\mathbf{y})\|^2 / (n-k)$, where $\hat{\boldsymbol{\mu}}_{\pi,i}(\mathbf{y}) = x_i \hat{\beta}_{\pi,i}(\mathbf{y})$.

Step 1: Inspired by Song and Cheng (2020), we define a test function

$$\phi(\mathbf{y}) = \mathbf{1}\{\, \|\hat{\boldsymbol{\mu}}_\pi(\mathbf{y}) - \boldsymbol{\mu}^*\| \geq \sqrt{n}\sigma^* \varepsilon_n \text{ and } \left|\hat{\sigma}^2_\pi(\mathbf{y}) - \sigma^{*2}\right| > \sigma^{*2}\varepsilon_n$$
$$\text{for some } \pi_k \text{ nested in } \pi^* \text{ with } k \leq (1+\delta) g_n^*\}$$

for some fixed $\delta > 0$ chosen later. Let $\circ$ denote the Hadamard product of two vectors. We define

$$C_n = \left\{ (\boldsymbol{\beta}, \sigma) : \|\mathbf{x} \circ \boldsymbol{\beta} - \boldsymbol{\mu}^*\| \leq M_1 \sqrt{n} \sigma^* \varepsilon_n \text{ and } \frac{1-\varepsilon_n}{1+\varepsilon_n} < \sigma^2/\sigma^{*2} < \frac{1+\varepsilon_n}{1-\varepsilon_n} \right\}^c \backslash B_n,$$

and

$$B_n = \Big\{ (\boldsymbol{\beta}, \sigma) : \text{The partition underlying } \boldsymbol{\beta} \text{ has at least } \delta g_n^* \text{ clusters } \Big\}.$$

For any π_k nested in π^* with $k \leq (1+\delta) g_n^*$ and the jth cluster $\mathcal{C}_j$ in π_k, we have $\hat{\beta}_{(j)} \sim \mathrm{N}\left(\beta^*_{(j)}, \sigma^{*2} / \sum_{i \in \mathcal{C}_j} x_i^2\right)$, where $\beta^*_{(j)}$ is the true coefficient in $\mathcal{C}_j$, and thus $\sum_{i \in \mathcal{C}_j} (x_i \hat{\beta}_{(j)} - x_i \beta^*_{(j)})^2 \sim \sigma^{*2} \chi^2_1$. Hence, $\|\hat{\boldsymbol{\mu}}_\pi(\mathbf{y}) - \boldsymbol{\mu}^*\|^2 / \sigma^{*2} \sim \chi^2_k$.

We now bound the type-I error of the test function. Since $k = O(g_n^*) \prec n\varepsilon_n^2$ by Assumption (C3) and $\varepsilon_n \asymp (g_n^* \log n/n)^{1/2}$, from the concentration inequality for χ^2 distribution in Lemma A.2, we have

$$\begin{aligned}
&\mathbb{P}_{(\beta^*,\sigma^*)}\left(\|\hat{\boldsymbol{\mu}}_\pi(\mathbf{y}) - \boldsymbol{\mu}^*\| \geq \sqrt{n}\sigma^*\varepsilon_n, \left|\hat{\sigma}_\pi^2(\mathbf{y}) - \sigma^{*2}\right| > \sigma^{*2}\varepsilon_n\right) \\
\leq\, &\mathbb{P}(\chi_k^2 \geq n\varepsilon_n^2) \leq \exp\left(-c_1' n\varepsilon_n^2\right),
\end{aligned}$$

for some constant $c_1' > 0$. Therefore, using a union bound and the second part of Assumption (C3),

$$\mathbb{E}_{(\beta^*,\sigma^*)}\{\phi(\mathbf{y})\} \leq P_n \cdot \exp\left(-c_1' n\varepsilon_n^2\right) \leq \exp\left(-c_1 n\varepsilon_n^2\right), \tag{A.1}$$

for some constant $c_1 > 0$ and large $n\varepsilon_n^2/(g_n^* \log n)$.

Next we bound the type-II error. We rewrite

$$C_n = C_n^{(1)} \cup C_n^{(2)}$$

where

$$C_n^{(1)} = \left\{(\boldsymbol{\beta}, \sigma) : \|\mathbf{x} \circ \boldsymbol{\beta} - \boldsymbol{\mu}^*\| > M_1\sqrt{n}\sigma^*\varepsilon_n, \frac{\sigma^2}{\sigma^{*2}} < \frac{1+\varepsilon_n}{1-\varepsilon_n}\right\} \cap B_n^c$$

and

$$C_n^{(2)} = \left\{\sigma : \frac{\sigma^2}{\sigma^{*2}} \leq \frac{1-\varepsilon_n}{1+\varepsilon_n} \text{ or } \frac{\sigma^2}{\sigma^{*2}} \geq \frac{1+\varepsilon_n}{1-\varepsilon_n}\right\} \cap B_n^c.$$

For any $(\boldsymbol{\beta}, \sigma) \in C_n$, let π be the corresponding partition of $\boldsymbol{\beta}$ and $\mathcal{T}$ be a spanning tree inducing π. Define $\hat{\pi}$ to be the partition formed by removing the edges $\left\{(i,j) \in \mathcal{E}_\mathcal{T} : |\beta_i - \beta_j| > 0 \text{ or } |\beta_i^* - \beta_j^*| > 0\right\}$ from $\mathcal{T}$. Then $\hat{\pi}$ is nested in both π and π^*, and has no more than $(1+\delta)g_n^*$ clusters (this is due to the construction of B_n^c and g_n^*). For any $\boldsymbol{\beta} \in C_n^{(1)}$, we have

$$\begin{aligned}
&\mathbb{P}_{(\beta,\sigma)}\left(\|\hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y}) - \boldsymbol{\mu}^*\| \leq \sqrt{n}\sigma^*\varepsilon_n\right) \\
&= \mathbb{P}_{(\beta,\sigma)}\left(\|\hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y}) - \mathbf{x} \circ \boldsymbol{\beta}) + \mathbf{x} \circ \boldsymbol{\beta} - \boldsymbol{\mu}^*\| \leq \sqrt{n}\sigma^*\varepsilon_n\right) \\
&\leq \mathbb{P}_{(\beta,\sigma)}\left(\|\hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y}) - \mathbf{x} \circ \boldsymbol{\beta}\| \geq \|\boldsymbol{\mu}^* - \mathbf{x} \circ \boldsymbol{\beta}\| - \sqrt{n}\sigma^*\varepsilon_n\right) \\
&\leq \mathbb{P}_{(\beta,\sigma)}\left(\|\hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y}) - \mathbf{x} \circ \boldsymbol{\beta}\| \geq (M_1 - 1)\sqrt{n}\sigma^*\varepsilon_n\right),
\end{aligned}$$

where the last inequality is due to the fact that when $\boldsymbol{\beta} \in C_n^{(1)}$, $\|\boldsymbol{\mu}^* - \mathbf{x} \circ \boldsymbol{\beta}\| > M_1\sqrt{n}\sigma^*\varepsilon_n$. Note also that within each cluster $\mathcal{C}_j$ under $\hat{\pi}$, $\sum_{i\in\mathcal{C}_j}\left(\hat{\boldsymbol{\mu}}_{\hat{\pi},i}(\mathbf{y}) - x_i\beta_{(j)}\right)^2 = \frac{\left(\sum_{i\in\mathcal{C}_j} x_i\epsilon_i\right)^2}{\sum_{i\in\mathcal{C}_j} x_i^2} \sim \sigma^2\chi_1^2$, where $\beta_{(j)}$ is the value of $\boldsymbol{\beta}$ in $\mathcal{C}_j$, and hence $\|\hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y}) - \mathbf{x} \circ \boldsymbol{\beta}\|^2/\sigma^2 \sim \chi_{\hat{k}}^2$ under the true parameters $(\boldsymbol{\beta}, \sigma)$, where $\hat{k}$ is the number of clusters in $\hat{\pi}$. Therefore,

$$\begin{aligned}
\mathbb{P}_{(\beta,\sigma)}\left(\|\hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y}) - \boldsymbol{\mu}^*\| \leq \sqrt{n}\sigma^*\varepsilon_n\right) &\leq \mathbb{P}\left(\chi_{\hat{k}}^2 \geq \frac{1-\varepsilon_n}{1+\varepsilon_n}(M_1-1)^2 n\varepsilon_n^2\right) \\
&\leq \exp\left(-c_2'(M_1-1)^2 n\varepsilon_n^2\right) \qquad \text{(A.2)}
\end{aligned}$$

for large M_1 and some constant $c_2' > 0$.

Now consider $(\boldsymbol{\beta}, \sigma) \in C_n^{(2)}$. By the normality of $\mathbf{y}$ we have $\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y})\|^2 \sim \sigma^2\chi_{n-\hat{k}}^2$. Therefore, since $\sigma \in C_n^{(2)}$,

$$\begin{aligned}
&\mathbb{P}_{(\beta,\sigma)}\left(\left|\hat{\sigma}_{\hat{\pi}}^2(\mathbf{y}) - \sigma^{*2}\right| < \sigma^{*2}\varepsilon_n\right) \\
&= \mathbb{P}_{(\beta,\sigma)}\left(\left|\frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y})\|^2}{\sigma^{*2}(n-\hat{k})} - 1\right| < \varepsilon_n\right) \\
&= \mathbb{P}_{(\beta,\sigma)}\left((1-\varepsilon_n)\frac{\sigma^{*2}}{\sigma^2} < \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y})\|^2}{\sigma^2(n-\hat{k})} < (1+\varepsilon_n)\frac{\sigma^{*2}}{\sigma^2}\right) \\
&\leq \mathbb{P}_{(\beta,\sigma)}\left(\left|\frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{\hat{\pi}}(\mathbf{y})\|^2}{\sigma^2} - (n-\hat{k})\right| > (n-\hat{k})\varepsilon_n\right) \\
&\leq \mathbb{P}\left(\left|\chi_{n-\hat{k}}^2 - (n-\hat{k})\right| > (n-\hat{k})\varepsilon_n\right) \\
&\leq \exp\left(-c_2 n\varepsilon_n^2\right), \qquad \text{(A.3)}
\end{aligned}$$

for some constant $c_2 > 0$ and large n.

Combining (A.2) and (A.3), we obtain

$$\begin{aligned}\sup_{(\beta,\sigma)\in C_n} \mathbb{E}_{(\beta,\sigma)}\{1-\phi(y)\} &\leq \max\left\{\exp\left(-c_2'(M_1-1)^2 n\varepsilon_n^2\right), \exp\left(-c_2 n\varepsilon_n^2\right)\right\} \\ &\leq \exp\left(-c_2 n\varepsilon_n^2\right), \qquad \text{(A.4)}\end{aligned}$$

if M_1 is chosen to be large.

Step 2: Let $m(\mathbf{y})$ be the marginal likelihood, $f^*(\mathbf{y})$ be the true likelihood and $\boldsymbol{\epsilon} = \mathbf{y} - \mathbf{x}\circ\boldsymbol{\beta}^*$ be the vector of error terms.

We claim that, with probability $\mathbb{P}\left(\|\boldsymbol{\epsilon}\| \leq 2\sqrt{n}\sigma^*\right)$,

$$\tilde{H}_n := \left\{(\boldsymbol{\beta},\sigma) : \left\|\frac{1}{\sigma}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^*\right)\right\|_\infty \leq g_n^*\frac{\log n}{n}, 0 \leq \sigma^2 - \sigma^{*2} \leq \sigma^{*2} g_n^* \frac{\log n}{n}\right\} \subset H_n,$$

where H_n is defined as

$$\begin{aligned}H_n = \Bigg\{(\boldsymbol{\beta},\sigma) : \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{x}\circ\boldsymbol{\beta}^* - \mathbf{x}\circ\boldsymbol{\beta} + \boldsymbol{\epsilon}\right\|^2 + \frac{\|\boldsymbol{\epsilon}\|^2}{2\sigma^{*2}} - n\log\frac{\sigma}{\sigma^*}\right) \\ \geq \exp(-c_3' g_n^* \log n)\Bigg\}\end{aligned}$$

for some constant $c_3' > 0$. Thus,

$$\begin{aligned}\frac{m(\mathbf{y})}{f^*(\mathbf{y})} &\geq \int_{H_n} \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{x}\circ\boldsymbol{\beta}^* - \mathbf{x}\circ\boldsymbol{\beta} + \boldsymbol{\epsilon}\right\|^2 + \frac{\|\boldsymbol{\epsilon}\|^2}{2\sigma^{*2}} - n\log\frac{\sigma}{\sigma^*}\right) p\left(\boldsymbol{\beta},\sigma^2\right) d\boldsymbol{\beta} d\sigma^2 \\ &\geq \Pi(H_n)\cdot\exp\left(-c_3' g_n^* \log n\right) \geq \Pi(\tilde{H}_n)\cdot\exp\left(-c_3' g_n^*\log n\right). \qquad \text{(A.5)}\end{aligned}$$

To see the claim, write

$$\begin{aligned}
&\frac{1}{2\sigma^2}\left\|\mathbf{x}\circ\boldsymbol{\beta}^*-\mathbf{x}\circ\boldsymbol{\beta}+\boldsymbol{\epsilon}\right\|^2-\frac{\left\|\boldsymbol{\epsilon}\right\|^2}{2\sigma^{*2}}+n\log\frac{\sigma}{\sigma^*}\\
=&\underbrace{\frac{\left\|\mathbf{x}\circ\boldsymbol{\beta}^*-\mathbf{x}\circ\boldsymbol{\beta}\right\|^2}{2\sigma^2}}_{I}+\underbrace{\frac{\left(\mathbf{x}\circ\boldsymbol{\beta}^*-\mathbf{x}\circ\boldsymbol{\beta}\right)^{\mathsf{T}}\boldsymbol{\epsilon}}{\sigma^2}}_{II}\underbrace{-\left\|\boldsymbol{\epsilon}\right\|^2\left(\frac{1}{2\sigma^{*2}}-\frac{1}{2\sigma^2}\right)}_{\leq 0\text{ since }\sigma\geq\sigma^*}+\underbrace{\frac{n}{2}\log\frac{\sigma^2}{\sigma^{*2}}}_{III}.
\end{aligned}$$

Noticing that when $(\boldsymbol{\beta},\sigma)\in\tilde{H}_n$ and $\|\boldsymbol{\epsilon}\|\leq 2\sqrt{n}\sigma^*$, by Assumption (C1) and the first part of Assumption (C3) we have

$$I\leq\frac{M_0^2}{\sigma^2}\left\|\boldsymbol{\beta}-\boldsymbol{\beta}^*\right\|^2\leq nM_0^2\left\|\frac{\boldsymbol{\beta}-\boldsymbol{\beta}^*}{\sigma}\right\|_\infty^2\leq M_0^2\frac{g_n^{*2}(\log n)^2}{n}=O\left(g_n^*\log n\right),$$

by Hölder's inequality and $\sigma^*\leq\sigma$ we have

$$\begin{aligned}
II&\leq 2\left\|\frac{\mathbf{x}\circ\boldsymbol{\beta}^*-\mathbf{x}\circ\boldsymbol{\beta}}{\sigma}\right\|_\infty\cdot\left\|\boldsymbol{\epsilon}\right\|_1\cdot\frac{1}{\sigma^*}\leq\frac{2M_0}{\sigma^*}\left\|\frac{\boldsymbol{\beta}^*-\boldsymbol{\beta}}{\sigma}\right\|_\infty\cdot\sqrt{n}\left\|\boldsymbol{\epsilon}\right\|\\
&\leq\frac{2M_0}{\sigma^*}\left\|\frac{\boldsymbol{\beta}^*-\boldsymbol{\beta}}{\sigma}\right\|_\infty\cdot\sqrt{n}\cdot 2\sqrt{n}\sigma^*=O\left(g_n^*\log n\right),
\end{aligned}$$

and

$$III\leq\frac{n}{2}\frac{g_n^*\log n}{n}=O\left(g_n^*\log n\right).$$

The claim then follows.

Next we show the prior assigns sufficient probability mass to $\tilde{H}_n$. Notice that $\Pi(\tilde{H}_n)=\sum_{\mathcal{T}\in\mathbb{T}_n}\Pi(\tilde{H}_n\mid\mathcal{T})\Pi(\mathcal{T})\geq\min_{\mathcal{T}\in\mathbb{T}_n}\Pi(\tilde{H}_n\mid\mathcal{T})$, and for each $\mathcal{T}$, $\Pi(\tilde{H}_n|\mathcal{T})\geq\Pi(\pi^*_{\mathcal{T}}\mid\mathcal{T})\Pi(\tilde{H}_n\mid\pi^*_{\mathcal{T}})$, where $\pi^*_{\mathcal{T}}$ is the partition obtained by removing the edges in $G^*_{\mathcal{T}}$ from $\mathcal{T}$. The number of clusters in $\pi^*_{\mathcal{T}}$, denoted by $k^*_{\mathcal{T}}$, is upper bounded by g_n^*.

First consider $\Pi(\pi^{\mathcal{T}} \mid \mathcal{T}) = \Pi(k = k^*_{\mathcal{T}})\binom{n-1}{k^*_{\mathcal{T}}-1}^{-1}$. By Assumption (C4),

$$\begin{aligned}\log \Pi(k = k^*_{\mathcal{T}}) &\geq \log \frac{(1-c)^{g^*_n}}{\sum_{k=1}^{n}(1-c)^k} \\ &= (g^*_n - 1)\log(1-c) + \log c - \log\{1-(1-c)^n\} \\ &\geq -2\alpha g^*_n \log n. \qquad \text{(A.6)}\end{aligned}$$

In addition,

$$-\log\binom{n-1}{g^*_n - 1} \geq -g^*_n \log n. \qquad \text{(A.7)}$$

Now we consider

$$\begin{aligned}\Pi(\tilde{H}_n|\pi^*_{\mathcal{T}}) = \Pi\Big(\frac{1}{\sigma}|\beta_{(j)} - \beta^*_{(j)}| \leq \frac{g^*_n \log n}{n} \text{ for } j = 1, 2, \ldots, k^*_{\mathcal{T}}, \\ 0 \leq \sigma^2 - \sigma^{*2} \leq \sigma^{*2} g^*_n \frac{\log n}{n}\Big).\end{aligned}$$

Since the prior for $\beta_{(j)}$ is given by

$$\beta_{(j)} \mid \lambda, \sigma \overset{iid}{\sim} \mathrm{N}(0, \lambda^{-1}\sigma^2), \quad \lambda \sim \mathrm{Gamma}(c_0/2, d_0/2),$$

by Assumption (C2) and (C3) we have, conditional on $0 \le \sigma^2 - \sigma^{*2} \le \sigma^{*2} g_n^* \frac{\log n}{n}$,

$$\begin{aligned}
&\Pi\left(\frac{1}{\sigma}|\beta_{(j)} - \beta^*_{(j)}| \le \frac{g_n^* \log n}{n} \text{ for all } j = 1, 2, \dots, k^*_{\mathcal{T}} \;\middle|\; \sigma\right) \\
&= \int_0^\infty \prod_{j=1}^{k^*_{\mathcal{T}}} \Pi\left(\frac{1}{\sigma}|\beta_{(j)} - \beta^*_{(j)}| \le \frac{g_n^* \log n}{n} \;\middle|\; \lambda, \sigma\right) p(\lambda) d\lambda \\
&\ge \int_0^\infty \left(\frac{g_n^* \log n}{n}\right)^{k^*_{\mathcal{T}}} \left(\frac{\lambda}{2\pi}\right)^{k^*_{\mathcal{T}}/2} \exp\left(-\frac{k^*_{\mathcal{T}}}{2} \lambda Z^2\right) \cdot p(\lambda) d\lambda, \\
&\qquad \text{where } Z = \max_{1 \le j \le k^*_{\mathcal{T}}} \frac{|\beta^*_{(j)}|}{\sigma^*} + 1 = \max_{1 \le i \le n} \frac{|\beta^*_i|}{\sigma^*} + 1, \\
&\ge \tilde{c}_3 \cdot \left(\frac{g_n^* \log n}{n}\right)^{g_n^*} \Gamma\left(\frac{k^*_{\mathcal{T}} + c_0}{2}\right) \cdot \left[\frac{1}{2}\{d_0 + g_n^* Z^2\}\right]^{-(g_n^* + c_0)/2}, \\
&\qquad \text{where } \tilde{c}_3 \text{ is a constant not involving } n, \\
&\ge \exp\left(-c_3'' g_n^* \log n\right) \qquad \text{(A.8)}
\end{aligned}$$

for some constant $c_3'' > 0$ when n is sufficiently large.

Finally, for some constant $c_3''' > 0$,

$$\begin{aligned}
&\Pi\left(0 \le \sigma^2 - \sigma^{*2} \le \sigma^{*2} g_n^* \frac{\log n}{n}\right) \\
&\ge \sigma^{*2} g_n^* \frac{\log n}{n} \cdot \min_{\sigma^2 \in [\sigma^{*2},\, \sigma^{*2}(1 + g_n^* \log n / n)]} p(\sigma^2) \\
&\ge \exp\left(-c_3''' g_n^* \log n\right). \qquad \text{(A.9)}
\end{aligned}$$

Combining (A.6), (A.7), (A.8) and (A.9) we obtain $\Pi(\tilde{H}_n \mid \mathcal{T}) \ge \exp(-c_3 g_n^* \log n)$ and thus $\Pi(\tilde{H}_n) \ge \exp(-c_3 g_n^* \log n)$, for some constant $c_3 > 0$ not depending on $\mathcal{T}$. Hence, with probability

$$\mathbb{P}\left(\|\boldsymbol{\epsilon}\| \le 2\sqrt{n}\sigma^*\right) \ge \mathbb{P}(\chi^2_n \le 4n) \ge 1 - \exp(-c_4 n), \qquad \text{(A.10)}$$

for some constant $c_4 > 0$, we have

$$\frac{m(\mathbf{y})}{f^*(\mathbf{y})} \ge \exp\left(-(c_3 + c_3') g_n^* \log n\right). \qquad \text{(A.11)}$$

Step 3: By Assumption (C4), for any $\mathcal{T}$ and some constant $c_5 > 0$ not depending on $\mathcal{T}$,

$$\begin{aligned}\Pi\left(B_n \mid \mathcal{T}\right) &\leq \Pi(k \geq \delta g_n^*)\\ &= \frac{\sum_{k=\delta g_n^*}^{n}(1-c)^k}{\sum_{k=1}^{n}(1-c)^k}\\ &= \frac{(1-c)^{\delta g_n^*-1}\{1-(1-c)^{n-\delta g_n^*+1}\}}{1-(1-c)^n}\\ &= O(1)\cdot(1-c)^{\delta g_n^*-1}\\ &\leq \exp\{-c_5\delta g_n^*\log n\}.\end{aligned}$$

We therefore have

$$\Pi\left(B_n\right) = \sum_{\mathcal{T}\in\mathbb{T}_n}\Pi\left(B_n \mid \mathcal{T}\right)\Pi(\mathcal{T}) \leq \exp\{-c_5\delta g_n^*\log n\}. \tag{A.12}$$

Combining parts: By Lemma A.3, (A.1), (A.4), (A.11), (A.12) and (A.10), it follows that for sufficiently large δ and $n\varepsilon_n^2/(g_n^*\log n)$,

$$\begin{aligned}&\mathbb{P}^*\left\{\Pi_n\left(\frac{1}{\sqrt{n}}\left\|\boldsymbol{\mu}-\boldsymbol{\mu}^*\right\| \geq M_1\sigma^*\varepsilon_n \mid \mathbf{y}\right) \geq \rho_n\right\}\\ &\leq \mathbb{P}^*\left\{\Pi_n\left(C_n\cup B_n \mid \mathbf{y}\right) \geq \rho_n\right\}\\ &\leq \exp(-g_n^*\log n)+\exp(-c_4 n)+\exp(-c_1 n\varepsilon_n^2),\end{aligned} \tag{A.13}$$

with

$$\rho_n = \frac{\exp\left(-c_2 n\varepsilon_n^2\right)+\exp\left(-c_5\delta g_n^*\log n\right)}{\exp(-g_n^*\log n)\exp\{-(c_3+c_3')g_n^*\log n\}} \to 0. \tag{A.14}$$

The result then follows from Borel-Cantelli lemma as the right-hand-side of (A.13) is summable. □

A.1.3 Proof of Propositon 2.5

We begin with the following lemmas.

Lemma A.4. *(Chernoff Bounds for Sum of Bernoulli Trials). Let* $z = \sum_{i=1}^{n} Z_i$*, where* $Z_i = 1$ *with probability* p_i *and* $Z_i = 0$ *with probability* $1 - p_i$*, and all* Z_i *are independent. Then* $\mathbb{P}\big(z \geq (1+\delta_2)\mathbb{E}(z)\big) \leq \exp\left(-\frac{\delta_2^2}{2+\delta_2}\mathbb{E}(z)\right) = \exp\left(-\frac{\delta_2^2}{2+\delta_2}\sum_{i=1}^{n} p_i\right)$*, for all* $\delta_2 > 0$*.*

Lemma A.5. *Under Assumption (C6), both the R-NN graph and the restricted Delaunay triangulation graph are connected graphs with probability 1 as* n *tends to infinity.*

Proof of Lemma A.5. By Theorem 1.1 in Penrose (1999), it is readily to check that the minimum value of the radius γ_1 such that R-NN is connected equals the maximum edge length of the MST on $\mathcal{S}_n$, and it scales with $\{(\pi p_s^{\min})^{-1} \log n/n\}^{1/2}$ with probability 1 as n tends to infinity. Notice that the MST is a subgraph of the Delaunay triangulation. By letting $\gamma_2 \asymp (\log n/n)^{1/2}$ and be larger than the maximum edge length of the minimum spanning tree, the restricted Delaunay triangulation contains all edges in the MST and hence is still a connected graph. □

Then we prove Proposition 2.5.

Proof of Proposition 2.5. Let $d(\mathbf{s}, \mathcal{B}) = \min_{\mathbf{s}_b \in \mathcal{B}} \|\mathbf{s} - \mathbf{s}_b\|$ denote the distance from a point $\mathbf{s} \in \mathbb{R}^2$ to a closed set $\mathcal{B} \subset \mathbb{R}^2$. For the boundary set $\mathcal{B}_{\beta^*}$, given $\upsilon_n > 0$, we define the υ_n-neighborhood of $\mathcal{B}_{\beta^*}$ as

$$\mathcal{N}(\mathcal{B}_{\beta^*}, \upsilon_n) = \left\{\mathbf{s} \in \mathbb{R}^2 : d(\mathbf{s}, \mathcal{B}_{\beta^*}) < \upsilon_n\right\}.$$

When Assumption (C6) holds, the maximum edge lengths in the R-NN graph and the restricted Delaunay triangulation graph scale with $(\log n/n)^{1/2}$. Therefore, by letting $\upsilon_n \asymp (\log n/n)^{1/2}$ and $\upsilon_n \geq \max(\gamma_1, \gamma_2)$, we can show that for any edge crossing $\mathcal{B}_{\beta^*}$, both of its endpoints must fall within $\mathcal{N}(\mathcal{B}_{\beta^*}, \upsilon_n)$.

We then define a set of edges from the original graph that have both endpoints within

υ_n distance to the boundary set $\mathcal{B}_{\beta^*}$ as follows

$$\mathcal{E}_{\mathcal{B}}(\upsilon_n) := \big\{(i,j) : (i,j) \in \mathcal{E}_0 \text{ and } \max\left\{d\left(\mathbf{s}_i, \mathcal{B}_{\beta^*}\right), d\left(\mathbf{s}_j, \mathcal{B}_{\beta^*}\right)\right\} \leq \upsilon_n\big\}.$$

From the Definition 2.4, it is readily to check that the edge differences are all zero when $(i,j) \in \mathcal{E}_0 \backslash \mathcal{E}_{\mathcal{B}}(\upsilon_n)$, i.e., $\sum\limits_{(i,j)\in\{\mathcal{E}_0\backslash\mathcal{E}_{\mathcal{B}}(\upsilon_n)\}} \left\|\beta_i^* - \beta_j^*\right\|_0 = 0$, where $\|\cdot\|_0$ is the L_0-norm.

For any given spanning tree $\mathcal{T}$, $|\mathcal{E}_{\mathcal{B}}(\upsilon_n) \cap \mathcal{E}_T| < z$, where $z = |\mathcal{S}_n \cap \mathcal{N}(\mathcal{B}_{\beta^*}, \upsilon_n)|$ denotes the number vertices falling within $\mathcal{N}(\mathcal{B}_{\beta^*}, \upsilon_n)$. The last inequality holds because $\mathcal{E}_{\mathcal{B}}(\upsilon_n) \cap \mathcal{E}_T$ is a spanning forest and hence its total number of edges is less than z.

Recall the boundary set $\mathcal{B}_{\beta^*}$ has a υ_n-covering number $N(\mathcal{B}_{\beta^*}, \upsilon_n, \|\cdot\|) \leq M_2\upsilon_n^{-1}$, it follows that the υ_n-packing number $M(\mathcal{B}_{\beta^*}, \upsilon_n, \|\cdot\|) \leq M_2\upsilon_n^{-1}$. From triangular inequality, there exists a maximal υ_n-packing for $\mathcal{B}_{\beta^*}$, denoted as $\mathbf{s}_{c,1}, \cdots, \mathbf{s}_{c,k}$ with the packing number $k \leq M_2\upsilon_n^{-1}$ such that

$$\bigcup_{j=1,\ldots,k} B\left(\mathbf{s}_{c,j}, \upsilon_n/2\right) \subset \mathcal{N}(\mathcal{B}_{\beta^*}, \upsilon_n) \subset \bigcup_{j=1,\ldots,k} B\left(\mathbf{s}_{c,j}, 2\upsilon_n\right) \tag{A.15}$$

where $B\left(\mathbf{s}_c, \upsilon_n\right)$ denotes a ball centered at $\mathbf{s}_c$ with radius υ_n.

Therefore, z follows a binomial distribution with size n and

$$\begin{aligned}
\mathbb{E}(z) &\leq \mathbb{E}\big(|\mathcal{S}_n \cap \{\bigcup_{j=1,\ldots,k} B\left(\mathbf{s}_{c,j}, 2\upsilon_n\right)\}|\big) \\
&\leq nk\mathbb{E}\big(|\mathbf{s}_i \cap B\left(\mathbf{s}_{c,j}, 2\upsilon_n\right)|\big) \\
&= nk \int_{B(\mathbf{s}_{c,j}, 2\upsilon_n)\cap[0,1]^2} p_s(\mathbf{s})d\mathbf{s} \\
&\leq 4\pi nk\upsilon_n^2 p_s^{\max} := E_{max} = O(nk\upsilon_n^2).
\end{aligned}$$

Let $\tilde{z}$ be another binomial distribution that is independent from z with size n and $\mathbb{E}(\tilde{z}) =$

E_{max}. From Lemma A.4,

$$\mathbb{P}\{z \geq (1+\delta_2)E_{max}\} \leq \mathbb{P}\{\tilde{z} \geq (1+\delta_2)E_{max}\} \leq \exp\left(-\frac{\delta_2^2}{2+\delta_2}E_{max}\right) \tag{A.16}$$

for all $\delta_2 > 0$. When $\upsilon_n \asymp (\log n/n)^{1/2}$, $E_{max} = O(nk\upsilon_n^2) = O(n\upsilon_n) = O\big\{(n\log n)^{1/2}\big\}$. Let $\delta_2 = 1$, then $P(z \geq 2E_{max}) \leq \exp\left(-E_{\max}/3\right) = \exp\{-M_4(n\log n)^{1/2}\}$ for some constant $M_4 > 0$. It implies with probability going to 1, the number of vertices falling within $\mathcal{N}(\mathcal{B}_{\beta^*}, \upsilon_n)$ is $O\{(n\log n)^{1/2}\}$.

Finally we have

$$\begin{aligned} |G_{\mathcal{T}}^*| = \sum_{(i,j)\in\mathcal{E}_T} \|\beta_i^* - \beta_j^*\|_0 &= \sum_{(i,j)\in\mathcal{E}_{\mathcal{B}}(\upsilon_n)\cap\mathcal{E}_T} \|\beta_i^* - \beta_j^*\|_0 + \sum_{(i,j)\in\{\mathcal{E}_0\setminus\mathcal{E}_{\mathcal{B}}(\upsilon_n)\}\cap\mathcal{E}_T} \|\beta_i^* - \beta_j^*\|_0 \\ &\leq |\mathcal{E}_{\mathcal{B}}(\upsilon_n)\cap\mathcal{E}_T| + \sum_{(i,j)\in\mathcal{E}_0\setminus\mathcal{E}_{\mathcal{B}}(\upsilon_n)} \|\beta_i^* - \beta_j^*\|_0 < z. \end{aligned}$$

Since z does not depend on the choice of $\mathcal{T}$, we have $g_n^* = \max_{\mathcal{T}\in\mathbb{T}_n} |G_{\mathcal{T}}^*| < z$. Combining with the result in (A.16), we complete the proof. □

A.1.4 Proof of Corollary 2.6

Proof. For $\mathcal{S}_n$ satisfying $g_n^* \leq M_3(n\log n)^{1/2}$ and $\log\tilde{P}_n \leq M_5 n^{1/2}\log^{3/2} n$, following the same proof of Theorem 2.3 with g_n^*, P_n and ε_n replaced by $M_3(n\log n)^{1/2}$, $\tilde{P}_n$ and $\tilde{\varepsilon}_n$ respectively, we have

$$\begin{aligned} &\mathbb{P}^*\left\{\Pi_n\left(\frac{1}{\sqrt{n}}\|\boldsymbol{\mu} - \boldsymbol{\mu}^*\| \geq M_6\sigma^*\tilde{\varepsilon}_n \mid \mathbf{y}, \mathcal{S}_n\right) \geq \rho_n \mid \mathcal{S}_n\right\} \\ &\leq \exp(-M_3 n^{1/2}\log^{3/2} n) + \exp(-c_4 n) + \exp(-c_1 n\tilde{\varepsilon}_n^2), \end{aligned}$$

where ρ_n has the same form as (A.14), with possibly different constants that do not depend on $\mathcal{S}_n$. Let Q_n be the event that $g_n^* \leq M_3(n\log n)^{1/2}$ and $\log\tilde{P}_n \leq M_5 n^{1/2}\log^{3/2} n$ hold.

Then

$$
\begin{aligned}
&\mathbb{P}^*\left\{\Pi_n\left(\frac{1}{\sqrt{n}}\|\boldsymbol{\mu}-\boldsymbol{\mu}^*\| \geq M_6\sigma^*\tilde{\varepsilon}_n \mid \mathbf{y}, \mathcal{S}_n\right) \leq \rho_n\right\} \\
\geq &\int_{Q_n} \mathbb{P}^*\left\{\Pi_n\left(\frac{1}{\sqrt{n}}\|\boldsymbol{\mu}-\boldsymbol{\mu}^*\| \geq M_6\sigma^*\tilde{\varepsilon}_n \mid \mathbf{y}, \mathcal{S}_n\right) \leq \rho_n \mid \mathcal{S}_n\right\} p_s(\mathcal{S}_n)\mathrm{d}\mathcal{S}_n \\
\geq &\left\{1-\exp(-M_3 n^{1/2}\log^{3/2} n)-\exp(-c_4 n)-\exp(-c_1 n\tilde{\varepsilon}_n^2)\right\}\cdot\mathbb{P}(Q_n).
\end{aligned}
$$

The result then follows since $\mathbb{P}(Q_n)\to 1$ and $\rho_n \to 0$ as n tends to infinity. □

A.1.5 Proof of Propositon 2.7

We begin with a brief review of Prim's algorithm for finding the MST and set up some notations. Prim's algorithm starts with an arbitrary vertex $\mathbf{s}_0$ of $\mathcal{G}_0$. In the t-th iteration, let $\mathcal{T}^t=(\mathcal{V}^t,\mathcal{E}^t)$ be a connected subgraph of the MST and $\tilde{\mathcal{E}}(\mathcal{V}^t)\subset\mathcal{E}_0$ be the set of all edges in $\mathcal{E}_0$ that has *one and only one* endpoint in $\mathcal{V}^t$ (for $t=0$, we define $\mathcal{T}^0=(\{\mathbf{s}_0\},\emptyset)$). $\mathcal{T}^t$ is constructed by picking the edge in $\tilde{\mathcal{E}}(\mathcal{V}^{t-1})$ with the least edge weight and adding this edge and its endpoint that is not in $\mathcal{V}^{t-1}$ into $\mathcal{T}^{t-1}$. The algorithm stops when $\mathcal{V}^t$ includes all the vertices in $\mathcal{G}_0$.

Proof of Propositon 2.7. Let A^t be the event that $\mathcal{T}^t$ is a connected subgraph of $\mathcal{T}$. It suffices to show that A^t happens with nonzero probability for all t. Notice that by Prim's algorithm, $A^t\subset A^{t-1}$ and thus

$$\mathbb{P}(A^t)=\mathbb{P}(A^t|A^{t-1})\mathbb{P}(A^{t-1}). \tag{A.17}$$

Consider two cases: (i) all vertices in $\mathcal{V}_0\setminus\mathcal{V}^{t-1}$ have different cluster memberships than the ones in $\mathcal{V}^{t-1}$, and (ii) otherwise. For (i), let e be an arbitrary edge in $\tilde{\mathcal{E}}(\mathcal{V}^{t-1})$. Then

$$\mathbb{P}(A^t|A^{t-1})\geq\mathbb{P}(\{e \text{ has the minimal weight among } \tilde{\mathcal{E}}(\mathcal{V}^{t-1})\})>0. \tag{A.18}$$

The strict inequality is due to the i.i.d. Unif$(1/2,1)$ on the weights of $\tilde{\mathcal{E}}(\mathcal{V}^{t-1})$. For (ii), let e

be an edge in $\tilde{\mathcal{E}}(\mathcal{V}^{t-1})$ connecting two endpoints in the same cluster. Then (A.18) still holds due to the way that we sample edge weights. The proposition then follows by induction arguments on t using (A.17).

□

A.2 RJ-MCMC Algorithm

In this appendix we provide details of our RJ-MCMC algorithm.

Recall from Section 2.3.4 that in each iteration of RJ-MCMC, we further iterate through each covariate from $m = 1$ to p. In each inner iteration one of the following four moves, birth, death, change, and hyper, is performed with probabilities $r_B(k_m), r_D(k_m), r_C(k_m)$ and $r_H(k_m)$, respectively. We set $r_B(k) = r_D(k) = 0.425$ for $k \in \{2, 3, \ldots, n-1\}$, $r_B(k) = 0.85$ for $k = 1$, $r_D(k) = 0.85$ for $k = n$, $r_C(k) = 0.1$ and $r_H(k) = 0.05$ for $k \in \{1, \ldots, n\}$.

Detailed implementation as well as acceptance probability of each move are given as follows.

(a) *Birth* ($k_m \to k_m + 1$): Randomly choose one edge from $n - k_m$ edges in the spanning tree $\mathcal{T}^{(m)}$ that connect vertices belonging to a same cluster with equal probability. Suppose the chosen edge connects two endpoints $\mathbf{s}_i, \mathbf{s}_{i'} \in \mathcal{C}_j^{(m)}$ with $i < i'$. By removing this edge we split $\mathcal{C}_j^{(m)}$ into two connected components, one containing $\mathbf{s}_i$ and other containing $\mathbf{s}_{i'}$. We set the component containing $\mathbf{s}_{i'}$ to be a new cluster $\mathcal{C}_{k_m+1}^{(m)\star}$ and set the other one to be $\mathcal{C}_j^{(m)\star}$. We let $\mathcal{C}_l^{(m)\star} = \mathcal{C}_l^{(m)}$ for $l = 1, \ldots, j-1, j+1, \ldots, k_m$. By doing so we propose a new partition $\pi^{(m)\star}$.

The acceptance probability is

$$\alpha_1 = \min\{1,\ \mathcal{A} \cdot \mathcal{P} \cdot \mathcal{L}\}, \tag{A.19}$$

where

$$\mathcal{A} = \frac{k_m}{n - k_m} \cdot (1 - c)$$

is the prior ratio,

$$\mathcal{P} = \frac{r_D(k_m+1)}{r_B(k_m)} \cdot \frac{n-k_m}{k_m}$$

is the proposal ratio,

$$\mathcal{L} = \frac{p\left[\mathbf{y} \mid \pi^{(m)\star}, k_m+1, \mathcal{T}^{(m)}, \left\{\pi^{(l)}, k_l, \mathcal{T}^{(l)}\right\}_{l\neq m}, \sigma^2, \lambda\right]}{p\left[\mathbf{y} \mid \{\pi^{(m)}, k_m, \mathcal{T}^{(m)}\}_{m=1}^{p}, \sigma^2, \lambda\right]}$$

is the likelihood ratio whose numerator and denominator are given by (2.6).

(b) *Death* ($k_m+1 \to k_m$): Randomly choose one edge from k_m edges in the spanning tree $\mathcal{T}^{(m)}$ that connect different clusters with equal probability. Suppose the chosen edge connects two endpoints $\mathbf{s}_i \in \mathcal{C}_j^{(m)}$ and $\mathbf{s}_{i'} \in \mathcal{C}_{j'}^{(m)}$ with $i < i'$. We merge these two clusters to be $C_j^{(m)\star}$ and remove $\mathcal{C}_{j'}^{(m)}$. We set $\mathcal{C}_l^{(m)\star} = \mathcal{C}_l^{(m)}$ for $l < j'$, and $\mathcal{C}_l^{(m)\star} = \mathcal{C}_{l+1}^{(m)}$ for $l \geq j'$. Then we propose $\pi^{(m)\star}$. The acceptance probability is the reciprocal of the one in birth step, i.e., $1/\alpha_1$, where α_1 is given by (A.19).

(c) *Change* ($k_m \to k_m$): First perform a death step by merging $\mathcal{C}_{j_1}^{(m)}$ and $\mathcal{C}_{j_2}^{(m)}$ to be $\mathcal{C}_{j_1'}^{(m)\star}$, and then perform a birth step by splitting $\mathcal{C}_{j_3}^{(m)\star}$ to be $\mathcal{C}_{j_3}^{(m)\star\star}$ and $\mathcal{C}_k^{(m)\star\star}$. The acceptance probability is $\alpha_1 = \min\{1,\ \mathcal{A}\cdot\mathcal{P}\cdot\mathcal{L}\}$, where $\mathcal{A}=1$, $\mathcal{P}=1$, and

$$\mathcal{L} = \frac{p\left[\mathbf{y} \mid \pi^{(m)\star\star}, k_m, \mathcal{T}^{(m)}, \left\{\pi^{(l)}, k_l, T^{(l)}\right\}_{l\neq m}, \sigma^2, \lambda\right]}{p\left[\mathbf{y} \mid \{\pi^{(m)}, k_m, T^{(m)}\}_{m=1}^{p}, \sigma^2, \lambda\right]}.$$

(d) *Hyper*: In this step $\mathcal{T}^{(m)}, \sigma^2$ and λ are updated. We first update σ^2 by a Gibbs step:

$$\sigma^2 \sim \mathrm{IG}\left(\frac{n+a_0}{2},\ \frac{1}{2}[b_0 + \mathbf{y}^\mathsf{T}\mathbf{P}_\lambda^{-1}\mathbf{y}]\right).$$

To update $\mathbf{w}^{(m)}$ (and hence $\mathcal{T}^{(m)}$), a Metropolis-Hastings procedure is utilized. We first sample edge weights of the cross-cluster edges from i.i.d. Unif $(1/2, 1)$ and edge weights of those within-cluster edges from i.i.d. Unif $(0, 1/2)$. Then we propose a new spanning tree using Prim's algorithm based on the new weights. The proposed spanning tree is guaranteed to induce the current partition $\pi^{(m)}$ (Teixeira et al., 2015). Since the full conditional of $\mathbf{w}^{(m)}$ remains the same for the proposed weights, the acceptance probability is always 1.

Finally we update λ using a Metropolis-Hastings step with a symmetric random walk proposal. We propose $\lambda^\star$ by

$$\log \lambda^\star \sim \mathrm{N}(\log \lambda, \sigma^2_{MH}),$$

and the acceptance probability is $\alpha_1 = \min\{1,\ \mathcal{A} \cdot \mathcal{P} \cdot \mathcal{L} \cdot \lambda^\star/\lambda\}$, where

$$\mathcal{A} = \left(\frac{\lambda^\star}{\lambda}\right)^{c_0/2-1} \exp\{-d_0(\lambda^\star - \lambda)/2\}$$

is the prior ratio, $\mathcal{P} = 1$ is the proposal ratio, and

$$\mathcal{L} = \frac{p\left[\mathbf{y} \mid \{\pi^{(m)}, k_m, \mathcal{T}^{(m)}\}_{m=1}^p, \sigma^2, \lambda^\star\right]}{p\left[\mathbf{y} \mid \{\pi^{(m)}, k_m, \mathcal{T}^{(m)}\}_{m=1}^p, \sigma^2, \lambda\right]}$$

is the likelihood ratio.

A.3 Additional Simulation Results

In this appendix we provide results on additional simulation settings.

A.3.1 Sensitivity Analysis of c

We first examine how sensitive the results from BSCC model to α. We reconsider the 100 data sets with moderate spatial correlation that are used in the Simulation Studies section. We fit BSCC models with four candidates $\alpha \in \{0.0075, 0.0150, 0.1000, 0.3333\}$, which give $c = 0.05, 0.1, 0.5, 0.9$, respectively.

Figure A.1 shows MSEs for BSCC models under different candidate values of α (or equivalently, c). We can see in all settings BSCC outperforms SCC in terms of MSEs, and

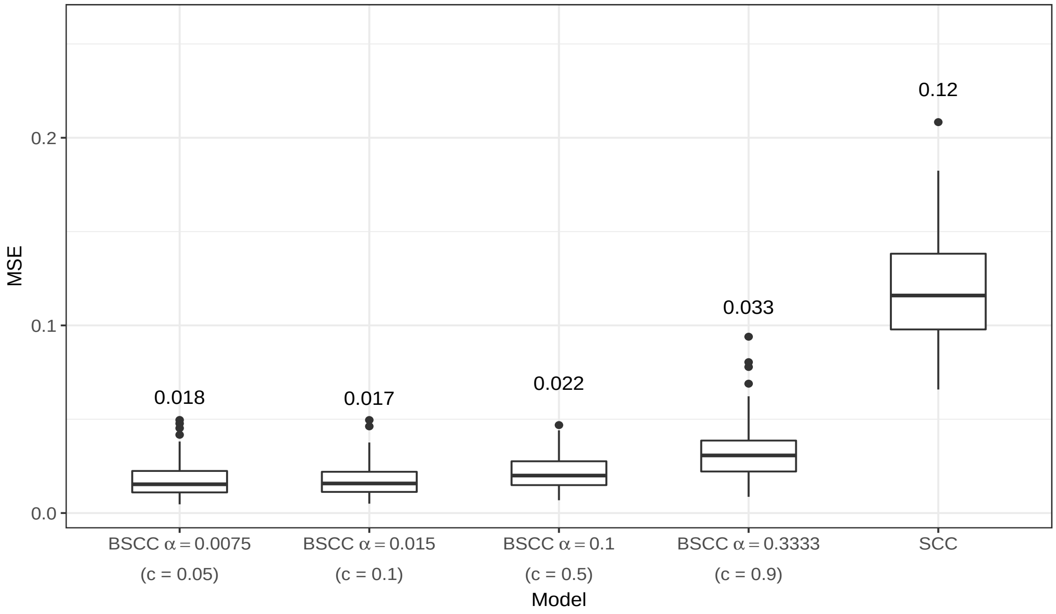

Figure A.1: Boxplots of MSEs for BSCC method under 4 different choices of hyperparameter α (or equivalently, c). 100 simulations are run for each choice. The average MSE_β over 100 simulations is shown above each box. MSEs for SCC method is also shown for reference.

overall the MSEs for BSCC are not sensitive to α (or c). However, careful choice of α does lead to improvements in MSEs.

Recall that Table 2.1 in the main text shows the number of data sets in which WAIC prefers a candidate value of α. In most of the data sets $\alpha = 0.0075$ or 0.0150 is preferred, which are two models with least MSE (see Figure A.1). Also notice that $\alpha = 0.3333$ that leads to higher MSE is rarely chosen by WAIC.

In summary, our simulation results suggest that the MSE performance is fairly robust to the choice of α (and thus c), as long as the value of α is within a reasonable range (e.g., $\alpha \leq 0.1$ in this example). We hence recommend using WAIC to determine the desired range of α.

A.3.2 Simulations under Different σ

In this subsection we evaluate the performance of BSCC under different settings of signal-to-noise ratio (SNR). We regenerate data sets from (2.12) with $\sigma \in \{0.1, 0.5, 0.75, 1\}$, and 100

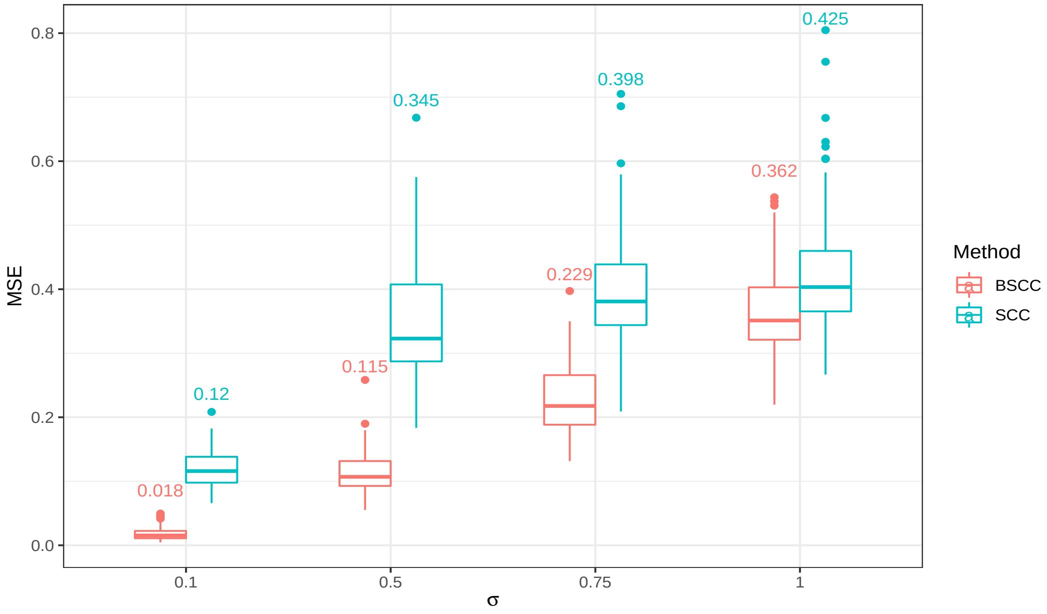

Figure A.2: Boxplots of MSEs for BSCC and SCC methods under 4 different choices of noise standard deviation σ. 100 simulations are run for each choice. The average MSE_β over 100 simulations is shown above each box.

data sets are generated for each value of σ. The rest data generating settings are the same as the ones for data sets with a moderate spatial correlation. The choices of σ correspond to different levels of SNR—as σ in increases, the variation in the residuals becomes larger with respect to spatially varying effects in $\mathbf{x}(\mathbf{s})\mathsf{T}\boldsymbol{\beta}(\mathbf{s})$. We fit BSCC and SCC models to each data set using the same settings as in the main text.

Figure A.2 presents boxplots of MSEs for both models under different choice of SNRs, and Table A.1 shows average Rand indices. As expected, the MSE performance of both methods degenerates as SNRs decrease. In terms of partition recovery, the Rand indices for BSCC also decreases as σ becomes larger. When $\sigma \in \{0.1, 0.5, 0.75\}$, BSCC outperforms SCC in both coefficient estimation and partition recovery. In the extreme case where $\sigma = 1$, BSCC still has a better MSE but slightly lower Rand indices.

	Rand index					
	β_1		β_2		β_3	
σ	BSCC	SCC	BSCC	SCC	BSCC	SCC
0.1	0.983	0.722	0.987	0.825	0.994	0.853
0.5	0.902	0.737	0.904	0.830	0.931	0.852
0.75	0.816	0.736	0.825	0.822	0.869	0.849
1	0.751	0.734	0.763	0.822	0.818	0.846

Table A.1: The average Rand indices for BSCC and SCC methods over 100 simulations under 4 different settings of SNR.

A.3.3 Simulations under Different Cross-Correlations

In many spatial applications, in addition to spatial dependence within each covariate, there may also be cross-dependence among covariates. In this subsection we investigate how BSCC performs under different settings of cross-dependence.

As discussed in Section 2.5.1, the two covariates in the simulation data are generated by a linear transformation of two independent Gaussian process realizations: $x_1(\mathbf{s}_i) = \zeta_1(\mathbf{s}_i), \quad x_2(\mathbf{s}_i) = r\zeta_1(\mathbf{s}_i) + \sqrt{1-r^2}\zeta_2(\mathbf{s}_i)$, where ζ_m $(m = 1, 2)$ is the realization of a Gaussian process and r controls the strength of cross-correlation between x_1 and x_2.

We consider $r \in \{0, 0.375, 0.75, 0.9\}$, which corresponds to zero, weak, moderate, and strong cross-correlation cases, respectively. For each value of r, we regenerate 100 data sets using the same true clustering patterns as Figure 2.2 in the main text shows. In practice, however, one may expect highly correlated covariates to have similar clustering configurations in their coefficients. As a result, we further consider a scenario where $r = 0.9$ and β_1 shares the same true partition as β_2 (Figure A.3). We refer to this scenario as "correlated partitions" in what follows. We fit BSCC and SCC models to each of them using the same settings as in the main text.

Figure A.4 shows MSEs under the five settings, and BSCC outperforms SCC in all of them. When β_1 and β_2 have different true clustering patterns, the MSE performance of BSCC is fairly robust to multicollinearity. This result is not surprising for two reasons.

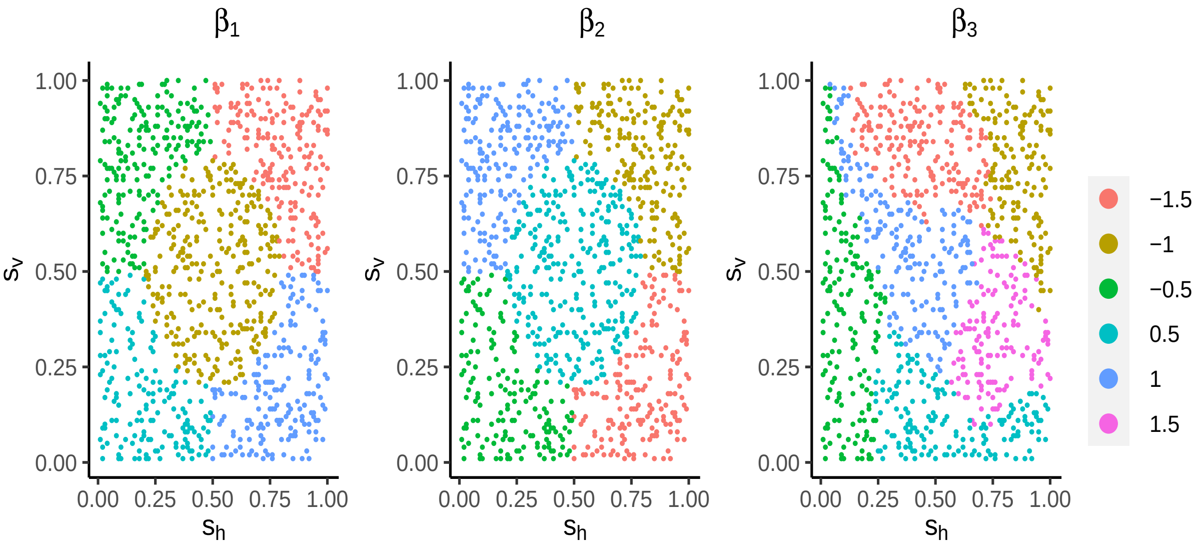

Figure A.3: Spatial structures of true coefficients used in the correlated partitions scenario in Section A.3.3, where β_1 and β_2 have the same true partitions.

First, we assume a ridge regression type of prior on $\boldsymbol{\beta}$ conditional on the partitions that mitigates multicollinearity problems. Second, the matrix $\tilde{\mathbf{X}}\top\tilde{\mathbf{X}}$ is well-conditioned when the partitions of β_1 and β_2 are different, where $\tilde{\mathbf{X}}$ is the transformed design matrix. When β_1 and β_2 share the same true partitions, the multicollinearity problem becomes more severe in $\tilde{\mathbf{X}}$ and we observe a drop in the accuracy of coefficient estimation.

The Rand indices under fives scenarios are shown in Table A.2. Similar to the findings in terms of MSEs, the partition estimation performance of BSCC is robust when β_1 and β_2 have different true partitions. On the other hand, when they have an identical partition, partition recovery for both coefficients become worse, probably due to the interference of the posterior distributions of the two partitions, as pointed out by an anonymous reviewer.

A.3.4 Comparisons with DPM Models with Spatial Random Effects

In this subsection we compare our method to the original version of the DPM model proposed by Ma et al. (2020), which includes a spatially varying intercept term modelled by a Gaussian process (referred to as DPM-GP model). We adopt the same hyperparameter settings as in the code provided in their paper, except that we set the maximum possible

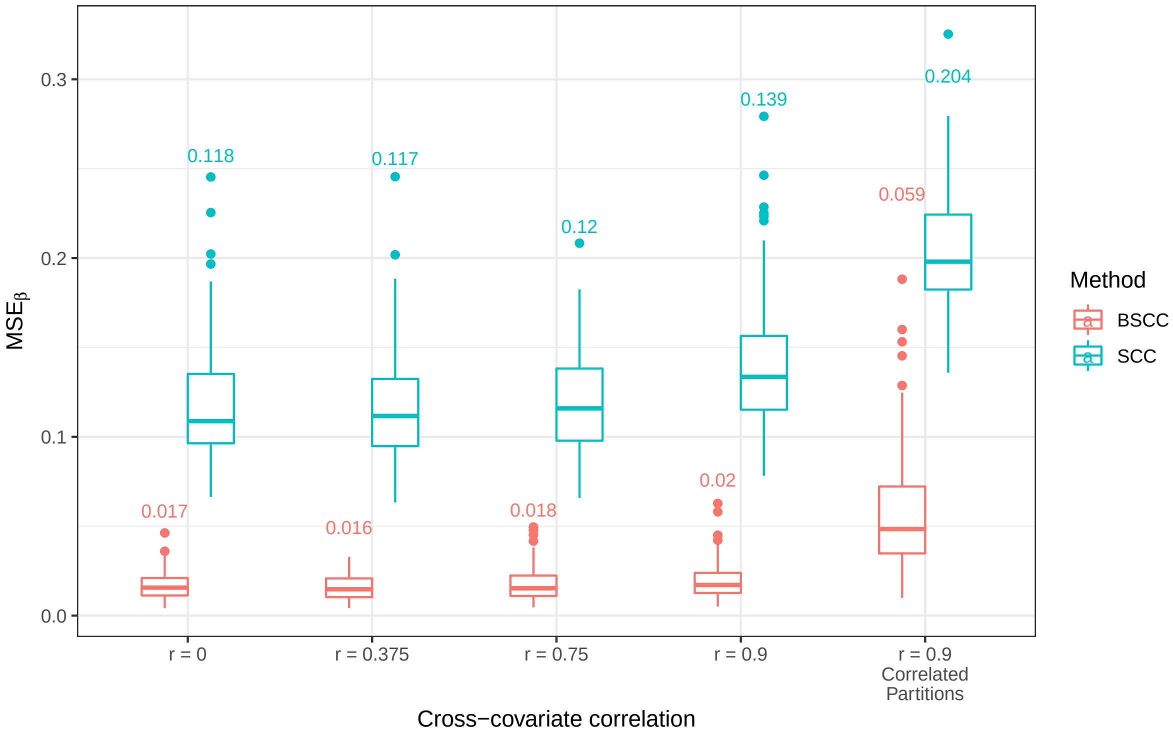

Figure A.4: Boxplots of MSEs for BSCC and SCC methods under 5 settings of cross-covariate correlation. "Correlated partitions" refers to the scenario where β_1 shares same true partition as β_2. 100 simulations are run for each choice. The average MSE_β over 100 simulations is shown above each box.

number of clusters to 50. We run the chain for 20,000 iterations, discard the first half, and collect posterior samples every 10 iterations after burn-in. It takes on average 11 hours to run a DPM-GP model for one simulation data set used in the main text. Due to its computational expensiveness, we only run the model for the first 10 data sets with a moderate spatial correlation.

Figure A.5 and Table A.3 show the MSEs and Rand indices of BSCC, SCC, DPM, and DPM-GP models for the 10 data sets, respectively. BSCC model achieves the best performance among the four models in estimating coefficient values and partitions.

	Rand index					
	β_1		β_2		β_3	
Cross-covariate correlation	BSCC	SCC	BSCC	SCC	BSCC	SCC
$r = 0$	0.984	0.719	0.988	0.824	0.994	0.853
$r = 0.375$	0.985	0.719	0.988	0.824	0.994	0.853
$r = 0.75$	0.983	0.722	0.987	0.825	0.994	0.853
$r = 0.9$	0.980	0.722	0.985	0.826	0.994	0.852
$r = 0.9$ with correlated partitions	0.961	0.830	0.963	0.829	0.989	0.853

Table A.2: The average Rand indices for BSCC and SCC methods over 100 simulations under 5 different settings of cross-covariate correlation.

	BSCC	SCC	DPM	DPM-GP
β_1	0.986	0.718	0.683	0.664
β_2	0.984	0.822	0.776	0.751
β_3	0.997	0.848	0.817	0.781

Table A.3: The average Rand indices for BSCC, SCC, DPM, and DPM-GP methods over 10 simulations with moderate spatial correlation.

A.4 Discussion on RJ-MCMC

A.4.1 Mixing of RJ-MCMC

In this subsection we discuss the mixing of tempered RJ-MCMC chains in more details. We consider the data set with a moderate spatial correlation that is analyzed in the Simulation Studies section of the main text, and compare the BSCC model fittings with and without parallel tempering (which are referred to as tempered and untempered models/chains, respectively, in what follows). Both chains are run for $50,000$ iterations after a burn-in period of the same length, and we thin the chains by taking samples every 20 iterations. For the tempered model, we adopt the sigmoidal temperature ladder (Gramacy and Taddy, 2010) with minimum inverse temperature $t_d = 0.35$ and run 8 parallel chains. See Section 2.5.1 in the main text for other settings of the RJ-MCMC algorithm.

Table A.4(a) shows acceptance rates of each move in each of the tempered chains. The

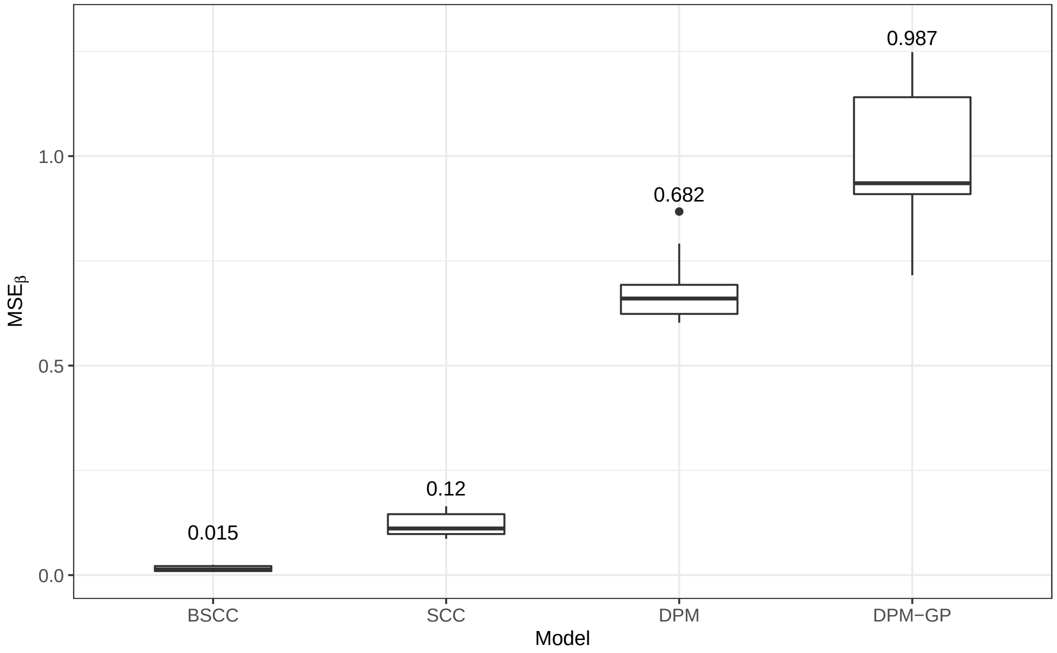

Figure A.5: Boxplots of MSEs for BSCC, SCC, DPM, and DPM-GP methods for 10 data sets with moderate spatial correlation. The average MSE_β over 10 simulations is shown above each box.

chains with inverse temperatures less than 1 have flatter target distributions than the posterior distribution, allowing for a more efficient exploration of the state space, as suggested by the fact that most of the chains with low inverse temperatures have higher Metropolis-Hastings acceptance rates. In particular, the acceptance rates for the Birth, Death, and Change moves of the hottest chain (i.e., with the lowest inverse temperature) are at least twice as high as their counterparts in the coolest chain.

Due to the higher acceptance rates, the hotter chains are able to visit the states that are hard to visit by conventional samplers. These states are passed to cooler chains via state swapping between chains. Acceptance rates of the swap attempts are shown in Table A.5. The swap acceptance rates are lower for hotter chains, probably due to larger gaps between adjacent inverse temperatures.

As a comparison, the acceptance rates for Metropolis-Hastings moves of the untempered chain are lower (Table A.4(b)), suggesting that the parallel tempering techniques can improve

(a) Tempered model

Chain #	Inverse temperature	Birth	Death	Change	Hyper
1	1.000	0.177	0.179	0.090	0.495
2	0.989	0.211	0.211	0.116	0.543
3	0.967	0.276	0.277	0.184	0.554
4	0.922	0.184	0.187	0.096	0.486
5	0.841	0.169	0.172	0.084	0.489
6	0.708	0.174	0.176	0.083	0.508
7	0.532	0.239	0.241	0.140	0.526
8	0.350	0.364	0.364	0.264	0.548

(b) Untempered model

Birth	Death	Change	Hyper
0.154	0.156	0.067	0.481

Table A.4: Acceptance rates of the four moves in (a) tempered model and (b) untempered model.

Chain #	1	2	3	4	5	6	7	8
Inverse temperature	1.000	0.989	0.967	0.922	0.841	0.708	0.532	0.350
Acceptance rate	0.620	0.526	0.581	0.566	0.453	0.367	0.144	0.055

Table A.5: Swap acceptance rates of tempered chains.

the efficiency for exploring the posterior space.

Traceplots of the thinned posterior densities after burn-in of the tempered and untempered models are shown in Figure A.6, where the densities for the tempered model are computed based on the draws from the coolest chain. The chains from both models seem to converge, but the tempered chain exhibits better mixing and less autocorrelation. The tempered chain transits between high posterior regions and low posterior regions more quickly and it visits low posterior regions more frequently.

Finally, we look at posterior distributions of the number of clusters for each coefficient obtained from the tempered and untempered models, which are shown in Figure A.7. The conventional untempered chain concentrates more on the regions near the posterior mode, while with the aid of parallel tempering, the tempered chain is able to visit some partitions

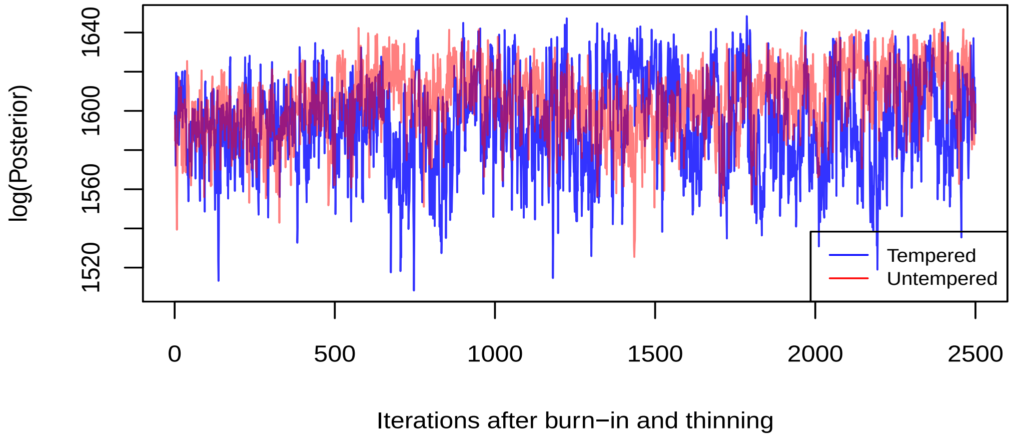

Figure A.6: Traceplot of thinned log posterior densities from tempered and untempered model after burn-in period.

that the untempered chain never does. For the coefficient β_3, for example, the tempered chain frequently visits partitions with 6 clusters, which are missed by the untempered chain. As indicated by the right tails, the untempered chain also underestimates the probability of getting partitions with large number of clusters.

A.4.2 Boundary-Adjusted Proposals

In this subsection we include the results of applying boundary-adjusted proposals (BAPs) for splitting clusters. The idea is that proposals splitting a cluster near its boundary is more likely to be accepted, which might improve mixing. BAPs thus assign higher probability on removing edges near boundaries. However, we do not observe satisfying improvement in mixing for this proposal. We summarize our methods and numerical results below.

Given partitions of all covariates $\{\pi^{(m)}\}_{m=1}^{p}$, we divide the vertex set $\mathcal{V}$ into two subsets, namely, *internal* vertices and *boundary* vertices, using 3-nearest neighbors methods. Specifically, a vertex is an internal vertex if all of its 3 nearest neighbors have the same cluster memberships for all covariates; otherwise, we treat it as a boundary vertex. We further divide the edge set $\mathcal{E}$ into three subsets to distinguish which edges are on the boundaries of clusters that we should target at:

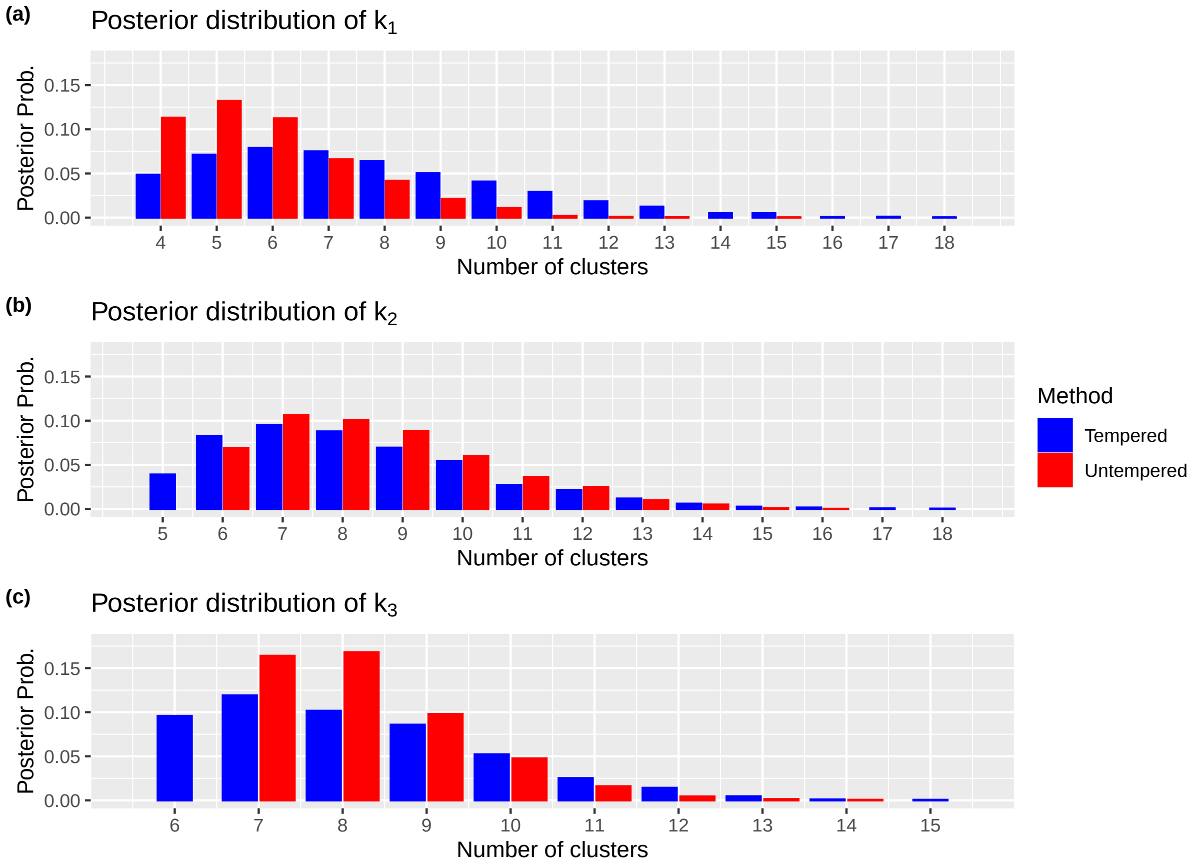

Figure A.7: Posterior distributions of k_m, the number of clusters for coefficient β_m, estimated from MCMC samples of the tempered and untempered models.

	Birth	Death	Change	Hyper
With BAP	0.142	0.145	0.068	0.477
Without BAP	0.154	0.156	0.067	0.481

Table A.6: Acceptance rates of the four moves with and without BAPs.

1. *Between-cluster* edges: We define an edge to be a *between-cluster* edge if it is connecting two vertices belonging to different clusters.

2. *Boundary* edges: We define an edge to be a *boundary* edge if it is not a between-cluster edge and at least one of its endpoints is a boundary vertex. BAPs place higher probability on removing this type of edges.

3. *Within-cluster* edges: We define an edge to be a *within-cluster* edge if it is not a between-cluster edge and both of its endpoints are internal vertices.

In BAPs, a cluster is uniformly chosen to be split. Then with probability p_w, a within-cluster edge that connects two vertices in this cluster is removed, and with probability $1-p_w$, a boundary edge is chosen to remove.

In this following simulation, we apply BAPs to the data set analyzed in Section A.4.1. We set $p_w = 0.2$ and do not apply parallel tempering.

Figure A.8 shows the thinned posterior densities after burn-in of the models with and without BAPs, and Table A.6 shows the acceptance rates of each move for both models. It seems that applying BAPs does not improve our results in terms of mixing and acceptance rates. Further investigations on more efficient partition proposals, including combining BAPs with parallel tempering, are left as future works.

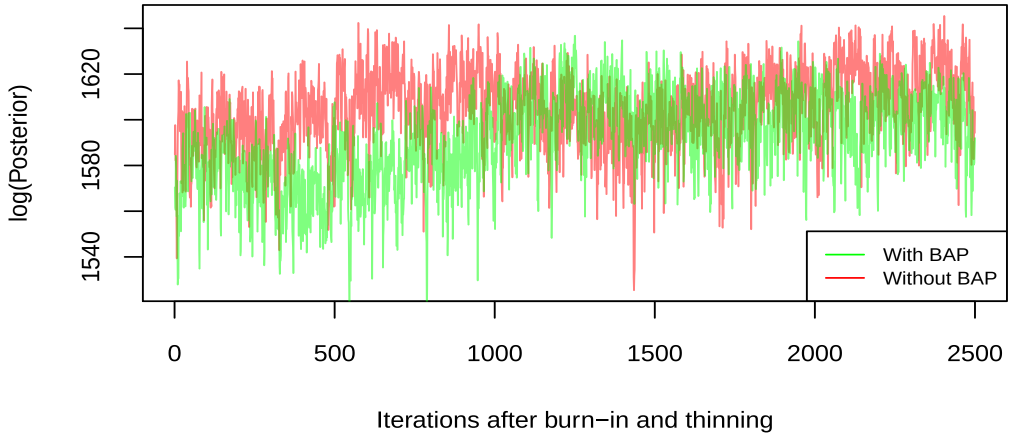

Figure A.8: Traceplot of thinned log posterior densities after burn-in period from models with and without BAPs.

APPENDIX B

SUPPLEMENTARY MATERIALS FOR CHAPTER 3

B.1 Proofs of Main Results

B.1.1 Kolmogorov Consistency of Soft Partitioned Gaussian Processes

Let $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}, m \in \{1, 2, \ldots\}$ be an arbitrary finite subset of $\mathcal{D}$. and $\mathbf{w}(\mathcal{V})$ be a random vector on $\mathcal{V}$ with the density $p(\mathbf{w}(\mathcal{V}))$ defined in (3.7). We will show that $p(\mathbf{w}(\mathcal{V}))$ satisfies Kolmogorov consistency conditions. We follow a similar proof as in Datta et al. (2016), but adaptations are needed to handle the extra soft partition term z in the conditional SPGP model formulation.

We start by showing that $p(\mathbf{w}(\mathcal{V}))$ is a proper density. In what follows, we use $\mathrm{d}(\mathbf{w}(\mathcal{V}))$ as a shorthand of $\prod_{\mathbf{v}_i \in \mathcal{V}} \mathrm{d}(\mathbf{w}(\mathbf{v}_i))$ in integrals. Note that the summation in (3.7) is over finite number of terms and thus the exchangeability of integration and summation is trivial. Let $\mathcal{U} = \mathcal{V} \setminus \mathcal{S}$. We have

$$
\begin{aligned}
&\int p(\mathbf{w}(\mathcal{V}))\,\mathrm{d}(\mathbf{w}(\mathcal{V})) = \int \left\{ \textstyle\sum_{\mathbf{z}(\mathcal{V})} p(\mathbf{w}(\mathcal{V})|\mathbf{z}(\mathcal{V}))p(\mathbf{z}(\mathcal{V})) \right\} \mathrm{d}(\mathbf{w}(\mathcal{V})) \\
&= \sum_{\mathbf{z}(\mathcal{V})} \int\int p(\mathbf{w}(\mathcal{U})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{U}))p(\mathbf{w}(\mathcal{S}))p(\mathbf{z}(\mathcal{V}))\,\mathrm{d}(\mathbf{w}(\mathcal{S} \setminus \mathcal{V}))\,\mathrm{d}(\mathbf{w}(\mathcal{V})) \\
&= \sum_{\mathbf{z}(\mathcal{V})} p(\mathbf{z}(\mathcal{V})) \int \left\{ \int p(\mathbf{w}(\mathcal{U})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{U}))\,\mathrm{d}(\mathbf{w}(\mathcal{U})) \right\} p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S})) \\
&= \sum_{\mathbf{z}(\mathcal{V})} p(\mathbf{z}(\mathcal{V})) \int p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S})) = \sum_{\mathbf{z}(\mathcal{V})} p(\mathbf{z}(\mathcal{V})) = 1,
\end{aligned}
$$

where the second equality uses (3.6) and the third one is due to $(\mathcal{S} \setminus \mathcal{V}) \cup \mathcal{V} = \mathcal{S} \cup \mathcal{U}$.

Let $\tau(1), \ldots, \tau(m)$ be an arbitrary permutation of $1, \ldots, m$. We now show that

$$
p(\mathbf{w}(\mathbf{v}_{\tau(1)}), \ldots, \mathbf{w}(\mathbf{v}_{\tau(m)})) = p(\mathbf{w}(\mathbf{v}_1), \ldots, \mathbf{w}(\mathbf{v}_m)).
$$

Since $\mathcal{S}$ is fixed and the ordering of locations in (3.6) and (3.7) does not matter, we know the density $p(\mathbf{w}(\mathcal{V}))$ is invariant under any permutation of locations in $\mathcal{V}$.

Next, we show that for any $\mathbf{v}_0 \in \mathcal{D}$, we have

$$p(\mathbf{w}(\mathcal{V})) = \int p(\mathbf{w}(\mathcal{V}_1))\,\mathrm{d}(w(\mathbf{v}_0)),$$

where $\mathcal{V}_1 = \mathcal{V} \cup \{\mathbf{v}_0\}$. We consider two cases. If $\mathbf{v}_0 \in \mathcal{S}$, then

$$\begin{aligned}
&\int p(\mathbf{w}(\mathcal{V}_1))\,\mathrm{d}(w(\mathbf{v}_0)) = \sum_{\mathbf{z}(\mathcal{V}_1)} p(\mathbf{z}(\mathcal{V}_1)) \int p(\mathbf{w}(\mathcal{V}_1)|\mathbf{z}(\mathcal{V}_1))\,\mathrm{d}(w(\mathbf{v}_0)) \\
&= \sum_{\mathbf{z}(\mathcal{V}_1)} p(\mathbf{z}(\mathcal{V}_1)) \int\int p(\mathbf{w}(\mathcal{V}_1 \setminus \mathcal{S})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{V}_1 \setminus \mathcal{S}))p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S} \setminus \mathcal{V}_1))\,\mathrm{d}(w(\mathbf{v}_0)) \\
&= \sum_{\mathbf{z}(\mathcal{V}_1)} p(\mathbf{z}(\mathcal{V}_1)) \int p(\mathbf{w}(\mathcal{V} \setminus \mathcal{S})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{V} \setminus \mathcal{S}))p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S} \setminus \mathcal{V})) \\
&= \sum_{\mathbf{z}(\mathcal{V}_1)} p(\mathbf{z}(\mathcal{V}_1))p(\mathbf{w}(\mathcal{V})|\mathbf{z}(\mathcal{V})) \\
&= \sum_{j=1}^{k} \left\{ \mathbb{P}(z(\mathbf{v}_0) = j) \sum_{\{\mathbf{z}(\mathcal{V}_1): z(\mathbf{v}_0)=j\}} p(\mathbf{z}(\mathcal{V}))p(\mathbf{w}(\mathcal{V})|\mathbf{z}(\mathcal{V})) \right\} \\
&= \sum_{j=1}^{k} \{\mathbb{P}(z(\mathbf{v}_0) = j)p(\mathbf{w}(\mathcal{V}))\} = p(\mathbf{w}(\mathcal{V})),
\end{aligned}$$

where we use the fact that $\mathcal{V}_1 \setminus \mathcal{S} = \mathcal{V} \setminus \mathcal{S}$, $(\mathcal{S} \setminus \mathcal{V}_1) \cup \{\mathbf{v}_0\} = \mathcal{S} \setminus \mathcal{V}$, and $z(\mathbf{v}_0)$ is independent from $\mathbf{z}(\mathcal{V})$ conditional on $\pi_k(\mathcal{S})$. In the other case where $\mathbf{v}_0 \notin \mathcal{S}$, using $\mathcal{S} \setminus \mathcal{V}_1 = \mathcal{S} \setminus \mathcal{V}$, we have

$$
\begin{aligned}
&\int p(\mathbf{w}(\mathcal{V}_1))\,\mathrm{d}(w(\mathbf{v}_0))\\
&=\sum_{\mathbf{z}(\mathcal{V}_1)} p(\mathbf{z}(\mathcal{V}_1)) \int\int p\left(\mathbf{w}(\mathcal{V}\setminus\mathcal{S}), w(\mathbf{v}_0)|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{V}\setminus\mathcal{S})\,, z(\mathbf{v}_0)\right)p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S}\setminus\mathcal{V}_1))\,\mathrm{d}(w(\mathbf{v}_0))\\
&=\sum_{\mathbf{z}(\mathcal{V}_1)} p(\mathbf{z}(\mathcal{V}_1)) \int\left\{\int p\left(\mathbf{w}(\mathcal{V}\setminus\mathcal{S}), w(\mathbf{v}_0)|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{V}\setminus\mathcal{S})\,, z(\mathbf{v}_0)\right)\mathrm{d}(w(\mathbf{v}_0))\right\}p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S}\setminus\mathcal{V}_1))\\
&=\sum_{\mathbf{z}(\mathcal{V})} p(\mathbf{z}(\mathcal{V})) \int\left\{\sum_{j=1}^{k} p\left(\mathbf{w}(\mathcal{V}\setminus\mathcal{S})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{V}\setminus\mathcal{S})\,, z(\mathbf{v}_0)=j\right)\mathbb{P}(z(\mathbf{v}_0)=j)\right\}p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S}\setminus\mathcal{V}))\\
&=\sum_{\mathbf{z}(\mathcal{V})} p(\mathbf{z}(\mathcal{V})) \int p\left(\mathbf{w}(\mathcal{V}\setminus\mathcal{S})|\mathbf{w}(\mathcal{S}), \mathbf{z}(\mathcal{V}\setminus\mathcal{S})\right)p(\mathbf{w}(\mathcal{S}))\,\mathrm{d}(\mathbf{w}(\mathcal{S}\setminus\mathcal{V}))\\
&=\sum_{\mathbf{z}(\mathcal{V})} p(\mathbf{z}(\mathcal{V}))p(\mathbf{w}(\mathcal{V})|\mathbf{z}(\mathcal{V})) = p(\mathbf{w}(\mathcal{V})).
\end{aligned}
$$

B.1.2 Proof of Propositions 3.3 and 3.4

We first formally define spatially contiguous partitions of a discrete set and a continuous domain, respectively (see, e.g., Castro et al. 2005; Luo et al. 2021b). For a generic set $\mathcal{A}$, we write its ϵ-covering numbers with respect to a norm $\|\cdot\|$ and a metric d as $N(\mathcal{B}, \epsilon, \|\cdot\|)$ and $N(\mathcal{B}, \epsilon, d)$, respectively.

Definition B.1. (i) Given an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ with a finite vertex set $\mathcal{S}$ and an edge set $\mathcal{E}$, we say $\pi_k(\mathcal{S}) = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ is a spatially contiguous partition of $\mathcal{S}$ with respect to $\mathcal{G}$ with k clusters if there exists a connected subgraph $\mathcal{G}_j = (\mathcal{S}_j, \mathcal{E}_j)$ of $\mathcal{G}$ for each $j = 1, \ldots, k$.

(ii) We say $\pi_k(\mathcal{D}) = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ is a spatially contiguous partition of a domain $\mathcal{D} \subseteq \mathbb{R}^d$ with k clusters if, there exists a boundary set $\mathcal{B} \subset \mathcal{D}$ such that $N(\mathcal{B}, v_n, \|\cdot\|_2) \leq c_0 v_n^{-(d-1)}$ for some constant $c_0 > 0$, and $\mathcal{D}_1 \setminus \mathcal{B}, \ldots, \mathcal{D}_k \setminus \mathcal{B}$ are the connected components of $\mathcal{D}$ in topological sense.

Our definition of spatially contiguous partitions of a discrete set is with respect to the notion of spatial graphs, whose edges encode the relationship of spatial adjacency or neighborhood.

Now we give proofs of Propositions 3.3 and 3.4.

Proofs of Propositions 3.3 and 3.4. Proposition 3.3 is a direct result combining Propositions 2 and 7 in Luo et al. (2021b). To prove Proposition 3.4, we follow Theorem 5.1 of Penrose (2007), which states that with probability one under the data generating process of $\mathcal{S}^*$,

$$\mathcal{L}\left\{\mathcal{D}_j \Delta \left(\cup_{\mathbf{s}^* \in \mathcal{S}^* \cap \mathcal{D}_j^*} V_{\mathbf{s}^*}\right)\right\} \longrightarrow 0 \quad \text{for } j = 1, \ldots, k,$$

since k is fixed. Defining $\pi_k(\mathcal{S}^*)$ such that $\mathcal{S}_j^* = \mathcal{S}^* \cap \mathcal{D}_j$, it follows from Proposition 3.3 that $\pi_k(\mathcal{S}^*)$ is within the support of the spanning-treed partition prior, which completes the proof. □

B.1.3 Lemmas

We first provide some lemmas that will be useful for proving Theorems 3.5 and 3.6. Throughout this and the following two subsections, we use the following notations. We write $\mathbf{y} = \mathbf{y}(\mathcal{S})$, $y_i = y(\mathbf{s}_i)$, and $\varepsilon_i = \varepsilon(\mathbf{s}_i)$, and omit the underlying set $\mathcal{A}$ in a partition $\pi(\mathcal{A})$ for conciseness. We also let $\bar{\tau}^* = \max_{1 \le j \le k^*} \tau_j^*$ and $\underline{\tau}^* = \min_{1 \le j \le k^*} \tau_j^*$. For a generic function $w(\cdot)$ defined on $[0,1]^2$, we denote $\|w\|_\infty = \sup_{\mathbf{s} \in [0,1]^2} |w(\mathbf{s})|$.

We let $p(y|\mathbf{s})$ be the density of y given $\mathbf{s}$ and $p^*(y|\mathbf{s})$ be the corresponding true one. Let $d_{\mathrm{TV}}(p_1, p_2) = \int |p_1(y|\mathbf{s}) - p_2(y|\mathbf{s})| \, p_s(\mathbf{s}) dy$ be the expected total variation distance between two densities $p_1(y|\mathbf{s})$ and $p_2(y|\mathbf{s})$ for the random design $\mathbf{s}$.

Lemma B.2. *(Lemma 1 of Laurent and Massart 2000) Let χ_d^2 be a chi-square distribution with degree of freedom d. Then the following concentration inequalities hold for any $x > 0$:*

$$\mathbb{P}\left(\chi_d^2 > d + 2x + 2\sqrt{dx}\right) \le \exp(-x)$$

and

$$\mathbb{P}\left(\chi_d^2 < d - 2\sqrt{dx}\right) \le \exp(-x).$$

Lemma B.3. *(Proposition 5 of Luo et al. 2021b) Let $\mathbb{T}_n$ be the set of all possible spanning trees of a spatial graph $\mathcal{G}$ with n vertices. Assume that $\mathcal{B}^*$ is the true boundary set and $\pi^*_{k^*_{\mathcal{T}}}$ is the partition induced by $\mathcal{T} \in \mathbb{T}_n$ if the edges of $\mathcal{T}$ across $\mathcal{B}^*$ are removed as in Section 3.3.3 (see the discussions before Assumption T and after Assumption SD). Let $k^*_{\mathcal{T}}$ be the number of clusters in $k^*_{\mathcal{T}}$. Under Assumption SD, there exist positive constants $c_1, \tilde{c}_1 > 0$, such that* $\max_{\mathcal{T}\in\mathbb{T}_n} k^*_{\mathcal{T}} \le c_1\sqrt{n\log n}$ *holds with probability at least* $1 - \exp\left(-\tilde{c}_1\sqrt{n\log n}\right)$.

The next lemma establishes an evidence lower bound for the STGP models. With some abuse of notations, we let $\Pi(\cdot)$ and $\Pi(\cdot|\mathcal{T})$ denote the marginal prior probability measure and the prior probability measure conditional on $\mathcal{T}$, respectively, such that $\Pi(\cdot) = \sum_{\mathcal{T}\in\mathbb{T}_n} \Pi(\cdot|\mathcal{T})\Pi(\mathcal{T})$, where $\Pi(\mathcal{T})$ is the marginal prior probability measure on $\mathcal{T}$.

Lemma B.4. *(Evidence lower bound) Under Assumptions T, SD. and P1, with probability at least* $1 - \exp(-4n) - \exp\left(-\tilde{c}_1\sqrt{n\log n}\right)$, *we have*

$$\int \prod_{i=1}^{n} \frac{p(y_i|\mathbf{s}_i)}{p^*(y_i|\mathbf{s}_i)} d\Pi(p) \ge \exp\left(-c_2 n\epsilon_n^2\right), \tag{B.1}$$

for some constants $c_2 > 0$ and $\tilde{c}_1$ as in Lemma B.3.

Proof of Lemma B.4. We will first show that (B.1) holds under the intersection of the events $E_1 = \{\sum_{i=1}^n \varepsilon_i^2/\tau^{*2}(\mathbf{s}_i) \le 4n, \text{for all } i = 1,\dots,n\}$ and $E_2 = \{\max_{\mathcal{T}\in\mathbb{T}_n} k^*_{\mathcal{T}} \le c_1\sqrt{n\log n}\}$, where c_1 is the constant in Lemma B.3. We will then show that the intersection of E_1 and E_2 holds with a large probability.

We proceed by first showing that under $E_1 \cap E_2$,

$$\int \prod_{i=1}^{n} \frac{p(y_i|\mathbf{s}_i)}{p^*(y_i|\mathbf{s}_i)} d\Pi(p|\mathcal{T}) \ge \exp\left(-c_2' n\epsilon_n^2\right)\Pi(N|\mathcal{T}), \tag{B.2}$$

where $c_2' = (\underline{\tau}^{*2} + 4\underline{\tau}^* + 1)/(2\underline{\tau}^{*2})$ and

$$N = \left\{ (\tilde{w}, \tau) : |\tilde{w}(\mathbf{s}_i) - \tilde{w}^*(\mathbf{s}_i)| < \epsilon_n^2,\ 1 \leq \frac{\tau^2(\mathbf{s}_i)}{\tau^{*2}(\mathbf{s}_i)} \leq 1 + \epsilon_n^2,\ \text{for all } i = 1, \ldots, n \right\}.$$

To show this, notice that

$$\begin{aligned}
&\int \prod_{i=1}^n \frac{p(y_i|\mathbf{s}_i)}{p^*(y_i|\mathbf{s}_i)} d\Pi(p|\mathcal{T}) \\
&\geq \int_N \prod_{i=1}^n \exp\left\{ \frac{1}{2}\log\frac{\tau^{*2}(\mathbf{s}_i)}{\tau^2(\mathbf{s}_i)} - \frac{(\tilde{w}^*(\mathbf{s}_i) - \tilde{w}(\mathbf{s}_i) + \varepsilon_i)^2}{2\tau^2(\mathbf{s}_i)} + \frac{\varepsilon_i^2}{2\tau^{*2}(\mathbf{s}_i)} \right\} d\Pi(\tilde{w}, \tau|\mathcal{T}) \\
&= \int_N \prod_{i=1}^n \exp\left\{ \frac{1}{2}\log\frac{\tau^{*2}(\mathbf{s}_i)}{\tau^2(\mathbf{s}_i)} - \frac{(\tilde{w}^*(\mathbf{s}_i) - \tilde{w}(\mathbf{s}_i))^2}{2\tau^2(\mathbf{s}_i)} - \frac{(\tilde{w}^*(\mathbf{s}_i) - \tilde{w}(\mathbf{s}_i))\varepsilon_i}{\tau^2(\mathbf{s}_i)} \right. \\
&\qquad\qquad\qquad\qquad \left. - \frac{\varepsilon_i^2}{2}\left(\frac{1}{\tau^2(\mathbf{s}_i)} - \frac{1}{\tau^{*2}(\mathbf{s}_i)} \right) \right\} d\Pi(\tilde{w}, \tau|\mathcal{T}) \\
&=: \int_N \prod_{i=1}^n \exp\left\{ \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV} \right\} d\Pi(\tilde{w}, \tau|\mathcal{T}).
\end{aligned}$$

Under N, $\mathrm{I} \geq -1/2 \cdot \log(1 + \epsilon_n^2) \geq -\epsilon_n^2/2$, $\mathrm{II} \geq -\epsilon_n^2/(2\underline{\tau}^{*2})$, and $\mathrm{IV} \geq 0$. Further under E_1, $\sum_{i=1}^n \mathrm{III} \geq -\sum_{i=1}^n |\tilde{w}^*(\mathbf{s}_i) - \tilde{w}(\mathbf{s}_i)| \cdot |\varepsilon_i/\tau(\mathbf{s}_i)| / \underline{\tau}^* \geq -(\epsilon_n^2/\underline{\tau}^*) \sum_{i=1}^n |\varepsilon_i/\tau(\mathbf{s}_i)| \geq -2n\epsilon_n^2/\underline{\tau}^*$ by using the inequality between L^1 and L^2 norms. (B.2) then follows.

Next we bound $\Pi(N|\mathcal{T})$ by considering the partition $\pi^*_{k^*_{\mathcal{T}}}$ obtained by removing all edges in $\mathcal{T}$ across the true boundary. Let

$$N' := \left\{ (\tilde{w}, \tau) : \left\| \tilde{w}_j - \tilde{w}_j^* \right\|_\infty < \epsilon_n^2,\ 1 \leq \frac{\tau_j^2}{\tau_j^{*2}} \leq 1 + \epsilon_n^2,\ \text{for each cluster } \mathcal{S}_j \text{ in } \pi^*_{k^*_{\mathcal{T}}} \right\} \subseteq N.$$

Then since $k^*_{\mathcal{T}} \leq \bar{k}_n$ under E_2 by Assumption P1, we have

$$
\begin{aligned}
&\Pi(N|\mathcal{T}) \geq \Pi(N'|\mathcal{T}) \\
&= \prod_{j=1}^{k^*_{\mathcal{T}}} \Pi_{\tilde{w}} \left(\left\| \tilde{w}_j - \tilde{w}^*_j \right\|_\infty < \epsilon_n^2 \right) \times \prod_{j=1}^{k^*_{\mathcal{T}}} \Pi_\tau \left(1 \leq \frac{\tau_j^2}{\tau_j^{*2}} \leq 1 + \epsilon_n^2 \right) \times \binom{n-1}{k^*_{\mathcal{T}}-1}^{-1} \times (1-c)^{k^*_{\mathcal{T}}} \\
&=: \mathrm{V} \times \mathrm{VI} \times \mathrm{VII} \times \mathrm{VIII}, \qquad \text{(B.3)}
\end{aligned}
$$

where $\Pi_{\tilde{w}}$ and Π_τ denote the prior probability measures on $\tilde{w}_j$ and τ_j^2, respectively.

Since $\pi^*_{k^*_{\mathcal{T}}}$ is nested in $\pi^*_{k^*}$, for a cluster $\mathcal{S}^*_j$ in $\pi^*_{k^*}$, we can write $\mathcal{S}^*_j = \mathcal{S}_{j_1} \cup \cdots \cup \mathcal{S}_{j_m}$ for some distinct clusters $\mathcal{S}_{j_1}, \ldots, \mathcal{S}_{j_m}$ in $\pi^*_{k^*_{\mathcal{T}}}$. Observe that $\mathcal{S}_{j_1}, \ldots, \mathcal{S}_{j_m}$ all share the same true mean function $\tilde{w}^*_j$ and true variance τ_j^{*2}. Recall that k^* is the fixed number of clusters in the true $\pi^*_{k^*}$. We obtain

$$
\begin{aligned}
\log \mathrm{V} &\geq k^*_{\mathcal{T}} \min_{1 \leq j \leq k^*_{\mathcal{T}}} \log \Pi_{\tilde{w}} \left(\left\| \tilde{w}_j - \tilde{w}^*_j \right\|_\infty < \epsilon_n^2 \right) \\
&= k^*_{\mathcal{T}} \min_{1 \leq j \leq k^*} \log \Pi_{\tilde{w}} \left(\left\| \tilde{w}_j - \tilde{w}^*_j \right\|_\infty < \epsilon_n^2 \right) \\
&\geq c_1 \sqrt{n \log n} \min_{1 \leq j \leq k^*} \log \Pi_{\tilde{w}} \left(\left\| \tilde{w}_j - \tilde{w}^*_j \right\|_\infty < \epsilon_n^2 \right),
\end{aligned}
$$

where the last inequality holds under event E_2. We further bound V using a similar argument as in the proof of Theorem 2 in Payne et al. (2020). Choosing σ_j and ϕ_j within a bounded neighborhood of the true σ^*_j and ϕ^*_j, respectively, then by Lemmas 3 and 4 in van der Vaart and van Zanten (2011) on the concentration function of a Matérn covariance GP prior, we have for some constant $\tilde{c}_2 > 0$,

$$
\log \mathrm{V} \geq -\tilde{c}_2 n \epsilon_n^2. \qquad \text{(B.4)}
$$

Since the prior Π_τ has a bounded density p_τ on $I_j := [\tau_j^{*2},\ \tau_j^{*2}(1+\epsilon_n^2)]$ for $j = 1, \ldots, k^*$,

$$\begin{aligned}
\log \mathrm{VI} &\geq k_{\mathcal{T}}^* \min_{1\leq j\leq k^*} \log \Pi_\tau\left(\tau_j^{*2} \leq \tau_j^2 \leq \tau_j^{*2}(1+\epsilon_n^2)\right) \\
&\geq k_{\mathcal{T}}^* \times \left\{ \log(\underline{\tau}^{*2}\epsilon_n^2) + \min_{1\leq j\leq k^*} \min_{\tau_j^2\in I_j} \log p_\tau(\tau_j^2) \right\} \\
&\geq c_1\sqrt{n\log n}(2\log\epsilon_n + \text{ constant}) \\
&\geq -\tilde{c}_2' n\epsilon_n^2, \qquad\qquad (\text{B.5})
\end{aligned}$$

for some constant $\tilde{c}_2' > 0$. Finally, for some constant $\tilde{c}_2'' > 0$,

$$\begin{aligned}
\log \mathrm{VII} + \log \mathrm{VIII} &\geq -k_{\mathcal{T}}^* \log n + k_{\mathcal{T}}^* \log(1-c) \\
&\geq -c_1\sqrt{n\log n}\,(\log n - \log(1-c)) \\
&\geq -\tilde{c}_2'' n\epsilon_n^2. \qquad\qquad (\text{B.6})
\end{aligned}$$

Combining (B.2)-(B.6), we have $\int \prod_{i=1}^n \frac{p(y_i,\mathbf{s}_i)}{p^*(y_i,\mathbf{s}_i)} d\Pi(p|\mathcal{T}) \geq \exp\left(-c_2 n\epsilon_n^2\right)$ with constant $c_2 = c_2' + \tilde{c}_2 + \tilde{c}_2' + \tilde{c}_2''$ under event $E_1 \cap E_2$. Hence,

$$\int \prod_{i=1}^n \frac{p(y_i|\mathbf{s}_i)}{p^*(y_i|\mathbf{s}_i)} d\Pi(p) = \sum_{\mathcal{T}\in\mathbb{T}_n} \Pi(\mathcal{T}) \cdot \int \frac{p(y_i|\mathbf{s}_i)}{p^*(y_i|\mathbf{s}_i)} d\Pi(p|\mathcal{T}) \geq \exp\left(-c_2 n\epsilon_n^2\right),$$

under $E_1 \cap E_2$, since c_2 does not depend on the choice of $\mathcal{T}$.

Finally, using Lemma B.2, $\mathbb{P}(E_1) \geq 1 - \exp(-4n)$. Combining with Lemma B.3, it is easy to show $\mathbb{P}(E_1 \cap E_2) \geq \mathbb{P}(E_1) + \mathbb{P}(E_2) - 1 \geq 1 - \exp(-4n) - \exp\left(-\tilde{c}_1\sqrt{n\log n}\right)$. This completes the proof.

□

Our last lemma is a useful tool for proving posterior concentration results by using test functions and evidence lower bounds.

Lemma B.5. *(Lemma A.3 of Song and Cheng 2020) Let $p \in \mathcal{P}_n$ be the likelihood func-*

tion with a prior Π *on a family of densities* $\mathcal{P}_n$*,* $p^* \in \mathcal{P}$ *be the true probability density of data generation,* $\mathbb{E}_p, \mathbb{E}^*$ *denote the expectations under* p *and* p^* *respectively,* $\mathbb{P}^*$ *denote the probability measure corresponding to the data generation density* p^**, and* $\Pi_n(\cdot|D_n)$ *denote the posterior given the data* D_n*. Let* B_n *and* C_n *be two subsets of the parameter space* $\mathcal{P}_n$*, and* φ_n *be a test function satisfying* $\varphi_n(D_n) \in [0,1]$ *for any data* D_n*. If* $\Pi(B_n) \leq b_n, \mathbb{E}^*\{\varphi_n(D_n)\} \leq b_n', \sup_{p \in C_n} \mathbb{E}_p\{1 - \varphi_n(D_n)\} \leq c_n$*, and*

$$\mathbb{P}^* \left(\int_{\mathcal{P}_n} \frac{p(D_n)}{p^*(D_n)} d\Pi(p) \geq a_n \right) \geq 1 - a_n'.$$

Then

$$\mathbb{E}^* \{\Pi_n (C_n \cup B_n \mid D_n)\} \leq \frac{b_n + c_n}{a_n} + a_n' + b_n'.$$

B.1.4 Proof of Theorem 3.5

Proof of Theorem 3.5. Note that the random spatial design has a bounded density $p_s(\mathbf{s})$ from Assumption SD. Also note that $\left|\int g(y|\mathbf{s})p(y|\mathbf{s})dy - \int g(y|\mathbf{s})p^*(y|\mathbf{s})dy\right| < \epsilon$ if and only if $\int g(y|\mathbf{s})p(y|\mathbf{s})dy - \int g(y|\mathbf{s})p^*(y|\mathbf{s})dy < \epsilon$ and $\int g(y|\mathbf{s})p^*(y|\mathbf{s})dy - \int g(y|\mathbf{s})p(y|\mathbf{s})dy = \int(1 - g(y|\mathbf{s}))p(y|\mathbf{s})dy - \int(1 - g(y|\mathbf{s}))p^*(y|\mathbf{s})dy < \epsilon$, it suffices to show the result holds for

$$\left\{ p : \int g(y|\mathbf{s})p^*(y|\mathbf{s})dy - \int g(y|\mathbf{s})p(y|\mathbf{s})dy < \epsilon \right\}, \tag{B.7}$$

for any bounded continuous function g. Redefine $W_{g,\epsilon}$ as the set in (B.7).

From Remark 4.4.1 of Ghosh and Ramamoorthi (2003), there exist test functions $\varphi_n(\mathbf{y}, \mathcal{S})$ such that $\mathbb{E}^*\{\varphi_n(\mathbf{y}, \mathcal{S})\} \leq c_3 n\epsilon$ and $\sup_{p \in W_{g,\epsilon}^c} \mathbb{E}_p\{1 - \varphi_n(\mathbf{y}, \mathcal{S})\} \leq c_3 n\epsilon$, for some constant $c_3 > 0$. Since $n\epsilon_n^2 = o(n)$, we have by Lemma B.4

$$\int \prod_{i=1}^n \frac{p(y_i|\mathbf{s}_i)}{p^*(y_i|\mathbf{s}_i)} d\Pi(p) \geq \exp(-c_3' n\epsilon),$$

for any constant $c_3' > 0$ and large enough n, with probability at least $1 - \exp(-4n) -$

$\exp\left(-\tilde{c}_1\sqrt{n\log n}\right)$. Choosing $c_3' < c_3$, by Lemma B.5,

$$\mathbb{E}^*\left\{\Pi_n(W_{g,\epsilon}^c|\mathbf{y},\mathcal{S})\right\} \le \exp\{-(c_3-c_3')n\epsilon\} + \exp(-4n) + \exp\left(-\tilde{c}_1\sqrt{n\log n}\right) + \exp(-c_3 n\epsilon).$$

It then follows from Markov inequality that, for any $\zeta > 0$,

$$\mathbb{P}^*\left\{\Pi_n(W_{g,\epsilon}^c|\mathbf{y},\mathcal{S}) > \zeta\right\} \le \frac{1}{\zeta}\left[\exp\{-(c_3-c_3')n\epsilon\} + \exp(-4n) + \exp\left(-\tilde{c}_1\sqrt{n\log n}\right) + \exp(-c_3 n\epsilon)\right].$$

We finish the proof by a direct application of Borel-Cantelli lemma. □

B.1.5 Proof of Theorem 3.6

Proof of Theorem 3.6. Note that the random spatial design has a bounded density $p_s(\mathbf{s})$ from Assumption SD. It suffices to show that Theorem 2 holds under a fixed spatial design.

Let $U_{M\epsilon_n} = \{p : d_{\mathrm{TV}}(p,p^*) < M\epsilon_n\}$ be an $M\epsilon_n$ total variation neighborhood of $p^*(y|\mathbf{s})$ for a large constant $M > 0$ chosen later. We proceed with three steps to verify the conditions in Lemma B.5.

Step 1: Sieve construction. For a generic function $w(s_1,s_2)$ defined on $\mathbb{R}^2$ and a vector $l = (l_1,l_2) \in \{0,1,2,\ldots\}^2$, we let $D^l w$ stand for $(\partial^{|l|}/\partial^{l_1}s_1\partial^{l_2}s_2)w(s_1,s_2)$, where $|l| = l_1 + l_2$. Consider a partition π_k of $\mathcal{S}$. For the jth cluster, let $C_{\pi_k,j} = C_{\pi_k,j}^w \times C_{\pi_k,j}^\tau$ be a subset of the parameter space of $(\tilde{w}_j, \tau_j^2)$, where

$$C_{\pi_k,j}^w = \left\{\tilde{w}_j : \left\|D^l\tilde{w}_j\right\|_\infty < M_n, |l| \le \alpha\right\}, \quad C_{\pi_k,j}^\tau = \left\{\tau_j^2 : a \le \tau_j^2 \le b\right\}.$$

Further let $C_{\pi_k} = \prod_{j=1}^k C_{\pi_k,j}$ be the product parameter space for a partition π_k, and C_n be the union of C_{π_k} for all possible spatially contiguous partitions π_k with $k \le \bar{k}_n$.

We now show that C_{π_k} satisfies the desired tail probability condition under the prior. Using Lemma 1 of Ghosal and Roy (2006), under Assumptions SD and P2, we have the conditional prior probability of C_{π_k} given $(\pi_k, k, \mathcal{T})$ satisfies $\Pi(C_{\pi_k}^c|\pi_k,k,\mathcal{T}) \le k\exp(-c_4 n\epsilon_n^2)$

for any constant $c_4 > 0$ and large enough n. Hence,

$$\begin{aligned}
\Pi(C_n^c|\mathcal{T}) &= \sum_{k=1}^{\bar{k}_n} \sum_{\pi_k} \Pi\left(C_{\pi_k}^c|\pi_k, k, \mathcal{T}\right) \cdot \binom{n-1}{k-1}^{-1} \cdot (1-c)^k \\
&\leq \sum_{k=1}^{\bar{k}_n} k \exp\left(-c_4 n \epsilon_n^2\right)(1-c)^k \\
&\leq \bar{k}_n^2 \exp\left(-c_4 n \epsilon_n^2\right) \leq \exp\left\{-(c_4-1)n\epsilon_n^2\right\},
\end{aligned}$$

where $\sum_{\pi_k}$ means summing over all possible spatially contiguous partitions with k clusters induced by $\mathcal{T}$, and by letting $B_n = U_{M\epsilon_n}^c \cap C_n^c$ we obtain

$$\Pi(B_n) \leq \sum_{\mathcal{T} \in \mathbb{T}_n} \Pi(C_n^c|\mathcal{T})\Pi(\mathcal{T}) \leq \exp\left\{-(c_4-1)n\epsilon_n^2\right\}. \tag{B.8}$$

Step 2: Existence of test functions. We verify the entropy conditions in Ghosal et al. (2000) for the existence of test functions.

Let $C_{\pi_k}^w = \prod_{j=1}^k C_{\pi_k,j}^w$ and $C_{\pi_k}^\tau = \prod_{j=1}^k C_{\pi_k,j}^\tau$. By Lemma 2 of Ghosal and Roy (2006), we have for some constant $c_5 > 0$ and any $\epsilon > 0$,

$$\log N\left(C_{\pi_k}^w, \epsilon, \|\cdot\|_\infty\right) \leq c_5 k \left(\frac{M_n}{\epsilon}\right)^{2/\alpha}.$$

We also cover $C_{\pi_k}^\tau$ by a $2\epsilon^2$-grid. Then the log covering number of $C_{\pi_k}^\tau$ with respect to the L^2 norm is

$$\log N\left(C_{\pi_k}^\tau, \epsilon, \|\cdot\|_2\right) = k \cdot \log \frac{b-a}{2\epsilon^2}.$$

With some abuse of notations we also let C_{π_k} denote the set of probability densities on $(y|\mathbf{s})$ determined by $(\tilde{w}_j, \tau_j^2) \in C_{\pi_k,j}^w \times C_{\pi_k,j}^\tau$ for $j = 1, \ldots, k$ under a partition π_k. Similarly, we let C_n be the union set of densities for all possible partitions.

Now we consider the covering number of C_{π_k}, viewed as a set of densities, under the total variation distance. We claim that, if for each cluster in π_k we have $\left\|\tilde{w}_j^{(1)} - \tilde{w}_j^{(2)}\right\|_\infty < 2\epsilon'$ and

$\left|\tau_j^{(1)2} - \tau_j^{(2)2}\right| < 2\epsilon'^2$, then $d_{\mathrm{TV}}\{p^{(1)}, p^{(2)}\} < c_5'\epsilon'$, for any $\epsilon' > 0$ and some constant $c_5' > 0$, where $p^{(i)}$ is the density on $(y|\mathbf{s})$ under $\{(\tilde{w}_j^{(i)}, \tau_j^{(i)})\}_{j=1:k} \in C_{\pi_k}$ for $i = 1, 2$.

To show the claim, using the KL divergence for Gaussian densities, we have for $\mathbf{s} \in \mathcal{S}_j$ under π_k,

$$\begin{aligned} K_{\mathbf{s}}\left(p^{(1)}, p^{(2)}\right) :&= \int p^{(1)}(y|\mathbf{s}) \log \frac{p^{(1)}(y|\mathbf{s})}{p^{(2)}(y|\mathbf{s})} dy \\ &= \frac{1}{2}\log\frac{\tau_j^{(2)2}}{\tau_j^{(1)2}} - \frac{1}{2}\left(1 - \frac{\tau_j^{(2)2}}{\tau_j^{(1)2}}\right) + \frac{1}{2}\frac{\left(\tilde{w}_j^{(1)}(\mathbf{s}) - \tilde{w}_j^{(2)}(\mathbf{s})\right)^2}{\tau_j^{(2)2}} \\ &\le \frac{1}{2}\log\left(1 + \frac{2\epsilon'^2}{a}\right) - \frac{1}{2} + \frac{1}{2}\left(1 + \frac{2\epsilon'^2}{a}\right) + \frac{1}{2}\frac{4\epsilon'^2}{a} \\ &\le \frac{4}{a}\epsilon'^2. \end{aligned}$$

Therefore, the KL divergence between $p^{(1)}$ and $p^{(2)}$ can be bounded as

$$\mathrm{KL}\left(p^{(1)}, p^{(2)}\right) = \int K_{\mathbf{s}}\left(p^{(1)}, p^{(2)}\right) p_s(\mathbf{s}) d\mathbf{s} \le \frac{4}{a}\epsilon'^2.$$

The claim then follows from

$$d_{\mathrm{TV}}\left\{p^{(1)}, p^{(2)}\right\} \le \sqrt{\frac{1}{2}\mathrm{KL}\left(p^{(1)}, p^{(2)}\right)} \le \sqrt{\frac{2}{a}}\epsilon'.$$

The claim suggests that

$$\log N\left(C_{\pi_k}, \epsilon_n, d_{\mathrm{TV}}\right) \le c_5'' k \left(\frac{M_n}{\epsilon_n}\right)^{2/\alpha} + k \log\frac{c_5'''}{\epsilon_n^2}$$

for some positive constants $c_5'', c_5''' > 0$, and thus

$$\log N\left(C_n, \epsilon_n, d_{\mathrm{TV}}\right) \le \log \bar{k}_n + \log\left(\max_{1 \le k \le \bar{k}_n} \xi(k)\right) + c_5'' \bar{k}_n \left(\frac{M_n}{\epsilon_n}\right)^{2/\alpha} + \bar{k}_n \log\frac{c_5'''}{\epsilon_n^2} \le \tilde{c}_5 n\epsilon_n^2$$

by Assumptions P1, (P2-2) and SG, for some constant $\tilde{c}_5 > 0$. Using Theorem 7.1 of Ghosal et al. (2000), there exist a test function $\varphi_n(\mathbf{y}, \mathcal{S})$ satisfying

$$\mathbb{E}^* \left\{\varphi_n(\mathbf{y}, \mathcal{S})\right\} \leq \exp\left(\tilde{c}_5 n\epsilon_n^2\right) \cdot \frac{\exp\left(-\tilde{c}_5' M^2 n\epsilon_n^2\right)}{1 - \exp\left(-\tilde{c}_5' M^2 n\epsilon_n^2\right)} \leq 2\exp\left\{-\left(\tilde{c}_5' M^2 - \tilde{c}_5\right) n\epsilon_n^2\right\}, \quad \text{(B.9)}$$

$$\sup_{p \in C_n \cap U_{M\epsilon_n}^c} \mathbb{E}_p \left\{\varphi_n(\mathbf{y}, \mathcal{S})\right\} \leq \exp\left(-\tilde{c}_5' M^2 n\epsilon_n^2\right), \quad \text{(B.10)}$$

for some constant $\tilde{c}_5' > 0$ and large n.

Step 3: Evidence lower bound. From Lemma B.4, we have (B.1) holds with probability at least $1 - \exp(-4n) - \exp\left(-\tilde{c}_1\sqrt{n\log n}\right)$, for some positive constants $\tilde{c}_1, c_2 > 0$.

Combining parts. Using Lemma B.5, we combine (B.8), (B.9), (B.10), and (B.1) to obtain that

$$\mathbb{E}^* \left\{\Pi_n\left(U_{M\epsilon_n}^c | \mathbf{y}, \mathcal{S}\right)\right\} \leq \frac{\exp\left\{-(c_4 - 1)n\epsilon_n^2\right\} + \exp\left(-\tilde{c}_5' M^2 n\epsilon_n^2\right)}{\exp\left(-c_2 n\epsilon_n^2\right)} + \exp(-4n) + \exp\left(-\tilde{c}_1\sqrt{n\log n}\right) + 2\exp\left\{-(\tilde{c}_5' M^2 - \tilde{c}_5)n\epsilon_n^2\right\}. \quad \text{(B.11)}$$

The result then follows from Markov inequality and Borel-Cantelli lemma if we choose c_4 and M such that $c_4 - 1 > c_2$ and $\tilde{c}_5' M^2 > \max(c_2, \tilde{c}_5)$.

□

B.2 Details on Posterior Inference

B.2.1 Estimation

In this subsection, we provide details on estimations of parameters and partitions. Recall that our MCMC algorithm proceeds as follows. Conditional on the global parameters $(\boldsymbol{\beta}, \lambda)$, the spanning-treed partitions $(\pi_k(\mathcal{S}), k, \mathcal{T})$ and the associated cluster-specified parameters $\left\{\tau_j^2, \bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j\right\}_{j=1:k}$ are updated via a reversible jump MCMC (RJ-MCMC) scheme. Then we update $\boldsymbol{\beta}$ and λ using Gibbs samplers conditional on $(\pi_k(\mathcal{S}), k, \mathcal{T})$ and $\left\{\tau_j^2, \bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j\right\}_{j=1:k}$.

To update the partitions and the associated covariance parameters, one of the birth, death, change, and hyper moves are randomly performed with probabilities $r_b(k) = 0.4$,

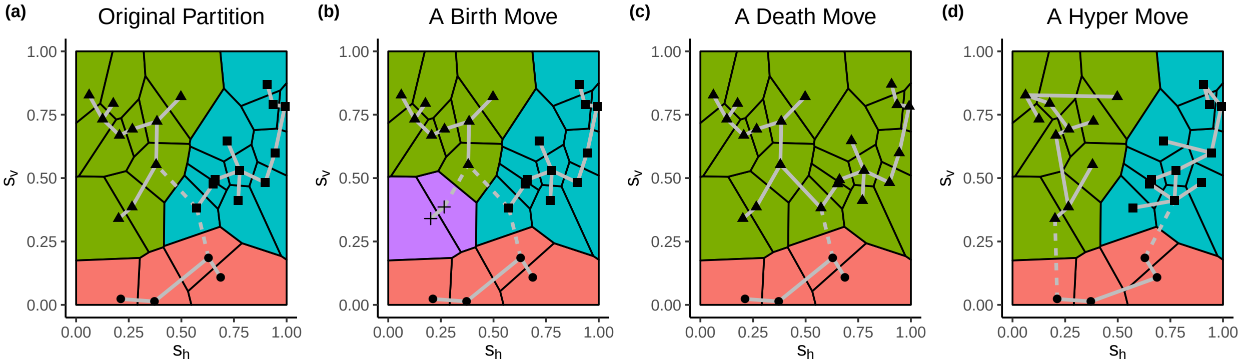

Figure B.1: Partitions and spanning trees obtained after (b) a birth, (c) a death, or (d) a hyper move from the original partition and tree in (a). Spanning tree edges across two distinct clusters are marked by dashed lines.

$r_d(k) = 0.4$, $r_c(k) = 0.19$, and $r_h(k) = 0.01$, respectively. Adjustments of the probabilities are made for $k = 1$ or $\bar{k}_m$. This combination of probabilities worked well in our experiments, but readers can modify them if desired. Each of the four moves adaptively updates the partitions or the spanning trees. Figure B.1 shows examples of a birth, a death, and a hyper move.

In the birth, death, and change moves, as discussed in Section 3.4.1, covariance parameters $(\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j)$ are updated by maximizing $p\left\{\mathbf{y}(\mathcal{S}_{j_2})|\bar{\sigma}^2, \tilde{\boldsymbol{\theta}}, -\right\} p(\bar{\sigma}^2)p(\tilde{\boldsymbol{\theta}})$, where the likelihood $p\left\{\mathbf{y}(\mathcal{S}_j)|\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j, -\right\}$ admits the form

$$p\left\{\mathbf{y}(\mathcal{S}_j)|\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j, -\right\} = \frac{b_\tau^{a_\tau}\Gamma(n_j/2 + a_\tau)}{(2\pi)^{n_j/2}\Gamma(a_\tau)} \left\{\det \tilde{\mathbf{C}}(\mathcal{S}_j|\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j)\right\}^{-1/2}$$
$$\times \left\{b_\tau + \frac{1}{2}\left(\mathbf{y}(\mathcal{S}_j) - \mathbf{X}(\mathcal{S}_j)\boldsymbol{\beta}\right)^{\mathsf{T}} \tilde{\mathbf{C}}(\mathcal{S}_j|\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j)^{-1} \left(\mathbf{y}(\mathcal{S}_j) - \mathbf{X}(\mathcal{S}_j)\boldsymbol{\beta}\right)\right\}^{-(n_j/2+a_\tau)}.$$

Here a_τ and b_τ are the shape and rate parameters for the inverse gamma prior on $\boldsymbol{\tau}^2$, respectively, and $\tilde{\mathbf{C}}(\mathcal{S}_j|\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j) = \bar{\sigma}_j^2\rho(\mathcal{S}_j, \mathcal{S}_j \mid \tilde{\boldsymbol{\theta}}_j) + \mathbf{I}_{n_j}$.

Conditional on $(\pi_k(\mathcal{S}), k, \mathcal{T})$ and $\left\{\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j\right\}_{j=1:k}$, we update the parameters $\{\tau_j^2\}_{j=1:k}$ by sampling from their full conditionals with the closed form

$$\tau_j^2|- \sim \mathrm{IG}\left(a_\tau + \frac{n_j}{2},\ b_\tau + \frac{1}{2}\left\{\mathbf{y}(\mathcal{S}_j) - \mathbf{X}(\mathcal{S}_j)\boldsymbol{\beta}\right\}^\mathsf{T} \tilde{\mathbf{C}}^{-1}(\mathcal{S}_j|\bar{\sigma}_j^2, \tilde{\boldsymbol{\theta}}_j)\left\{\mathbf{y}(\mathcal{S}_j) - \mathbf{X}(\mathcal{S}_j)\boldsymbol{\beta}\right\}\right).$$

Finally, we update the global parameters. Specifically, we sample $\boldsymbol{\beta}$ and λ from their full conditionals conditional on $(\boldsymbol{\Theta}, \pi_k, k, \mathcal{T})$. The updates take closed forms:

$$\boldsymbol{\beta}|- \sim \mathrm{N}_p(\mathbf{Q}_\beta^{-1}\mathbf{b}_\beta,\ \mathbf{Q}_\beta^{-1}),$$

with

$$\mathbf{Q}_\beta = \sum_{j=1}^{k} \mathbf{X}(\mathcal{S}_j)^\mathsf{T}\mathbf{C}^{-1}(\mathcal{S}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)\mathbf{X}(\mathcal{S}_j) + \lambda\mathbf{I}_p, \quad \mathbf{b}_\beta = \sum_{j=1}^{k} \mathbf{X}(\mathcal{S}_j)^\mathsf{T}\mathbf{C}^{-1}(\mathcal{S}_j, \mathcal{S}_j|\boldsymbol{\theta}_j)\mathbf{y}(\mathcal{S}_j),$$

and

$$\lambda|- \sim \mathrm{IG}\left(a_\lambda + \frac{p}{2},\ b_\lambda + \frac{\|\boldsymbol{\beta}\|_2^2}{2}\right).$$

B.2.2 Prediction

The algorithm for drawing posterior predictive samples is summarized Algorithm 2.

B.3 Supplementary Simulations

B.3.1 Kriging means and SDs

Figure B.2 illustrates the kriging means and standard deviations (SDs) across $\mathcal{D} = [0, 1]^2$ with various values of L and $\alpha_\ell = 1/L$ for $\ell = 1, \ldots, L$, given a partition $\pi_2(\mathcal{S}) = (\mathcal{S}_1, \mathcal{S}_2)$ and $\mathbf{w}(\mathcal{S})$ at a set of uniformly drawn locations $\mathcal{S}$. The partition π_2 divides $\mathcal{S}$ into two clusters: the one inside the grey circle and the other one outside the circle. $\mathbf{w}(\mathcal{S}_1)$ and $\mathbf{w}(\mathcal{S}_2)$ are realizations of two different stationary GPs. Note how the kriging means become smoother near the boundary as L increases. In the extreme case where $L = 1$, the kriging mean is discontinuous around the boundary (Panel (a)). The higher kriging SDs near the

Algorithm 2: Posterior predictive inference

```
Input : $\mathbf{X}(\mathcal{U})$, $\mathbf{y}(\mathcal{S})$, a posterior sample of $(\boldsymbol{\Theta}, \pi_k, k)$, number of neighbors $L$, and
        a vector of probabilities for neighbor choice $\boldsymbol{\alpha}$.
for $i \leftarrow 1$ to $r$ do
    Find the $\ell$th nearest neighbor of $\mathbf{u}_i$ in $\mathcal{S}$ for $\ell = 1, \dots, L$ ;
    Sample cluster membership $z(\mathbf{u}_i) \sim \text{Cat}\{z(N_{\mathbf{u}_i,1}), \dots, z(N_{\mathbf{u}_i,L}) | \alpha_1, \dots, \alpha_L\}$ ;
for $j \leftarrow 1$ to $k$ do
    Set $\mathcal{U}_j \leftarrow \{\mathbf{u} \in \mathcal{U} : z(\mathbf{u}) = j\}$ ;
    if $\mathcal{U}_j \neq \emptyset$ then
        Sample $\mathbf{y}(\mathcal{U}_j)$ from a multivariate Gaussian distribution with mean
          $\tilde{\boldsymbol{\mu}}(\mathcal{U}|\mathcal{S}, \boldsymbol{\theta}) = \mathbf{X}(\mathcal{U})\boldsymbol{\beta} + \mathbf{C}(\mathcal{U}, \mathcal{S}|\boldsymbol{\theta})\mathbf{C}^{-1}(\mathcal{S}, \mathcal{S}|\boldsymbol{\theta}) \{\mathbf{y}(\mathcal{S}) - \mathbf{X}(\mathcal{S})\boldsymbol{\beta}\}$ and
          covariance matrix
          $\boldsymbol{\Sigma}(\mathcal{U}|\mathcal{S}, \boldsymbol{\theta}) = \mathbf{C}(\mathcal{U}, \mathcal{U}|\boldsymbol{\theta}) - \mathbf{C}(\mathcal{U}, \mathcal{S}|\boldsymbol{\theta})\mathbf{C}^{-1}(\mathcal{S}, \mathcal{S}|\boldsymbol{\theta})\mathbf{C}(\mathcal{S}, \mathcal{U}|\boldsymbol{\theta})$ ;
Output: A posterior predictive sample $\mathbf{y}(\mathcal{U})$
```

circle when $L = 3$ (Panel (e)) and $L = 5$ (Panel (f)) indicate the high uncertainty due to the unknown cluster memberships of unobserved locations. In some applications, capturing abrupt changes is a desired feature, whereas in other applications, the random fields near the boundary can be relatively smooth across the partitions. An important implication of this example is that L-SPGP can be a flexible tool that accommodates both situations by choosing different L for modeling locally stationary spatial fields, as we will see in the following sections.

B.3.2 Additional results of isotropic processes

The true model of simulation in Section 3.5 has a constant true nugget effect SD $\tau^*(\mathbf{s}) \equiv 0.1$ and true $\tilde{w}(\mathbf{s}) \sim \text{GP}(\beta^*, \sigma_j^{*2}\rho_j^*)$ for $\mathbf{s} \in \mathcal{D}_j^*$, $j = 1, 2$. We set the global constant intercept as $\beta^* = 1$. The true ρ_j^* is taken to be an isotropic Matérn correlation function with $\nu = 5/2$. We set $\sigma_1^{*2} = 1$, $\sigma_2^{*2} = 0.5$, $\phi_1^* = 0.3$, and $\phi_2^* = 1$, so that the true processes in $\mathcal{D}_1^*$ and $\mathcal{D}_2^*$ have well-separated microergodic parameters $\vartheta = \sigma^2/\phi^{2\nu}$.

Regarding prior and other model choices, we construct a Delaunay triangulation graph, with edges longer than 0.1 removed. For the prior on k, we set $\bar{k}_n = 111 \approx 2\sqrt{n \log n}$ and $c = 0.5$. Priors for covariance parameters are specified as $\bar{\sigma}_j^2 \overset{\text{iid}}{\sim} \text{IG}(1, 10)$, $\tau_j^2 \overset{\text{iid}}{\sim} \text{IG}(2, 0.1)$,

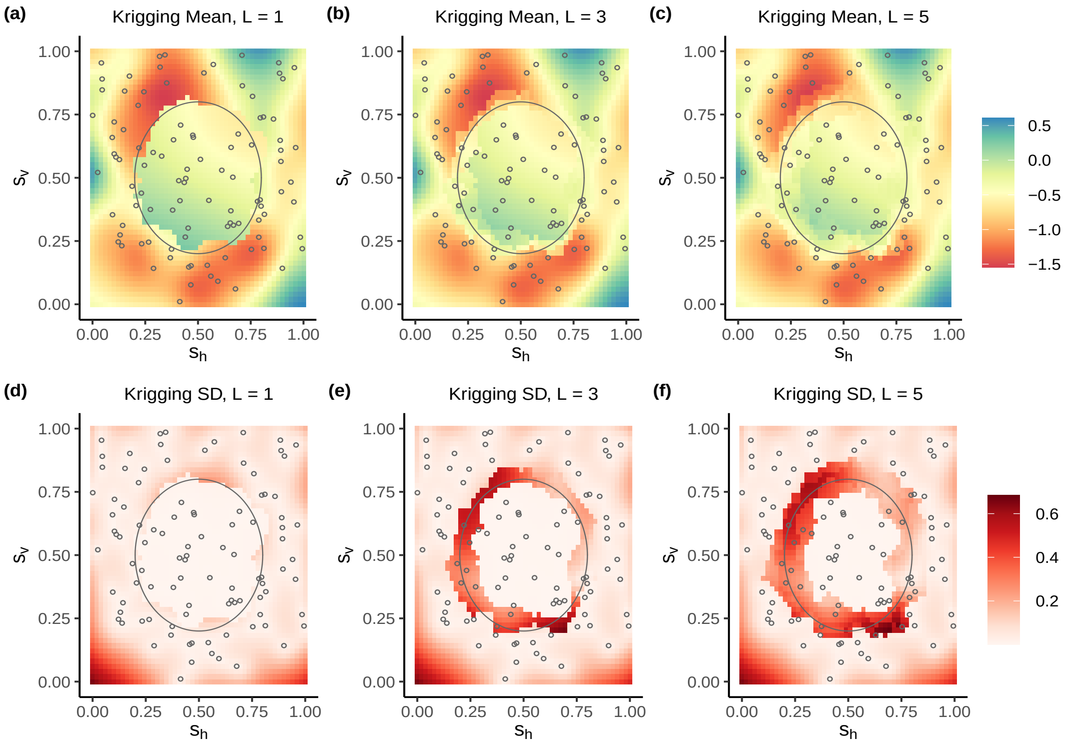

Figure B.2: Illustration of kriging mean and SDs for $L = 1, 3, 5$, given $\mathbf{w}(\mathcal{S})$ and a partition of $\mathcal{S}$ into two clusters separated by the grey circle. Locations in $\mathcal{S}$ are marked as black dots.

and $\phi_j \overset{\text{iid}}{\sim} t_1^+(5)$, where $t_{df}^+(s)$ stands for a half-t distribution with degree of freedom df and scale parameter s, and fix $\nu = 5/2$. We place an IG(2, 2) prior for λ, and specify $\boldsymbol{\mu}_\beta = 0$. As in Gramacy (2007) and Konomi et al. (2014), we restrict the minimum size of each cluster to be 30 such that covariance parameters can be well-estimated. For out-of-sample prediction, we use $L = 1$, 3, or 5 nearest neighbors to predict cluster memberships, and set the mixture weights (see Equation 3.9) as $\alpha_\ell = 1/L$ for $\ell = 1, \dots, L$.

For the competing models, we specify the covariance functions in the TGP and the SGP models as isotropic Matérn with $\nu = 5/2$. In the NSGP model, we set $\tau^2(\mathbf{s})$ to be a constant but allow for spatially varying variance and anisotropy covariance parameters, which are modeled by a reduced-rank GP with 36 equally-spaced knots and a linear function of spatial coordinates, respectively. In the SGP and NSGP models, the mean function is specified to be a constant with a Guassian prior; in TGP we adopt a cluster-wise constant mean function to which we assign a Gaussian prior with inverse Gamma variance (which is similar to the prior settings of STGP). All competing models are implemented in a Bayesian framework. We use the R package tgp (Gramacy, 2007) to fit TGP models. Both NSGP and SGP models are fitted by the R package BayesNSGP (Risser and Turek, 2020). Inferences are all based on exact likelihood (i.e., without likelihood or covariance approximations) for fair comparisons.

We show the plots of predictive densities at two selected locations in Figure B.3. The first chosen location is near the true boundary (called a boundary point). Its predictive density from STGP is bimodal, suggesting the uncertainty that it can be classified into either cluster near the true boundary since its neighbors do not belong to the same estimated cluster. The higher mode appears near the true value. This again confirms that STGP can quantify prediction uncertainty in a desirable way. The density from TGP is unimodal and its mode does not match the true value, possibly because this location is not close to the estimated boundary of TGP. The NSGP model gives a similar density as TGP does, except that its 95% HPD interval fails to cover the true value. Another location is selected to be an interior point, that is, a location not close to the true boundary, whose predictive density is expected to be unimodal as there is less uncertainty in cluster membership estimation. All three models perform well in prediction in the sense that the posterior densities are all unimodal with a mode near the true value.

B.3.3 Anisotropic processes

This study has a similar setup as in Section 3.5, except that the true data generating process in $\mathcal{D}_1^*$ is anisotropic and that we consider the anisotropic STGP models. The correlation function takes the form (Stein, 1999):

$$\rho(\mathbf{s},\mathbf{s}'|\tilde{\boldsymbol{\theta}}) = \frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\sqrt{2\nu}\, d(\mathbf{s},\mathbf{s}'|\tilde{\boldsymbol{\theta}})\right)^{\nu} K_{\nu}\left(\sqrt{2\nu}\, d(\mathbf{s},\mathbf{s}'|\tilde{\boldsymbol{\theta}})\right),$$

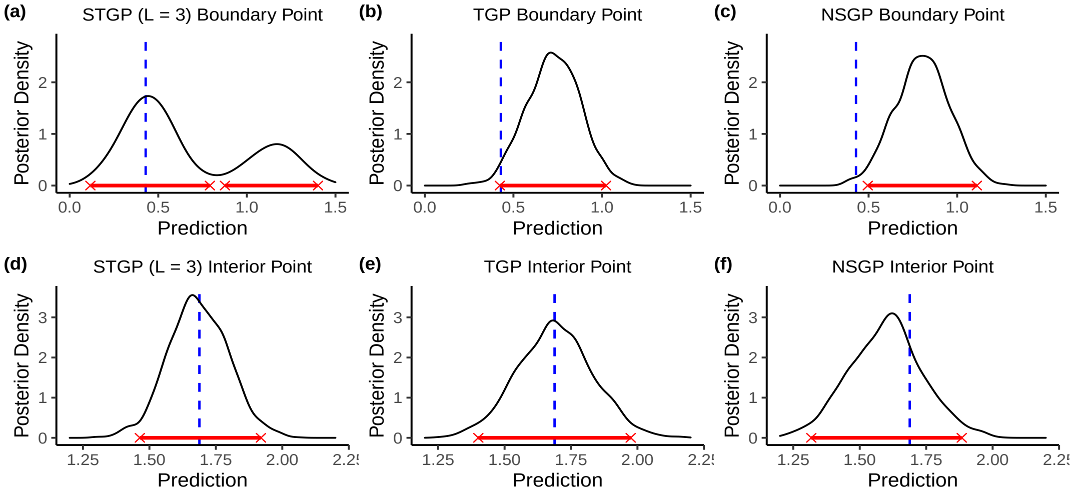

Figure B.3: Posterior predictive densities for a selected location near the true boundary (a-c) and a location in the interior of a true cluster (d-f). Blue dashed lines indicate the true values. 95% HPD intervals are marked by red segments.

with $d(\mathbf{s}, \mathbf{s}'|\phi_1, \phi_2, \psi) = (\mathbf{s} - \mathbf{s}')^\top \mathbf{\Psi}^\top \mathbf{D}^{-1} \mathbf{\Psi}(\mathbf{s} - \mathbf{s}')$, where $\mathbf{D} = \mathrm{diag}(\phi_1, \phi_2)$, $\mathbf{\Psi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}$ with $\phi_1, \phi_2 > 0$ and $0 \le \psi < \pi$.

More specifically, ρ_1^* and ρ_2^* are geometric Matérn correlation functions above with true parameters $\phi_{11}^* = 0.3$, $\phi_{21}^* = 1$, $\psi_1^* = \pi/4$, $\phi_{12}^* = \phi_{22}^* = 1$, and $\psi_2^* = 0$ (note that ρ_2^* is in fact isotropic). Other data generating setting is the same as in Section 3.5. The generated responses are visualized in Figure B.4(a).

We adopt the following priors for the covariance parameters in the anisotropic STGP models: conditional on the number of clusters k, we assign $\phi_{\ell j} \overset{\text{iid}}{\sim} t_1^+(5)$ and $\psi_j \overset{\text{iid}}{\sim} \mathrm{Unif}(0, \pi)$ for $\ell = 1, 2$ and $j = 1, \ldots, k$. Other settings on priors and spatial graphs are identical to those in Study 1. Since the `tgp` package (Gramacy, 2007) does not support geometric anisotropic kernels, we use a separable exponential correlation function for TGP instead. We also specify the covariance function in SGP as geometric anisotropic Matérn, while we employ the same NSGP model as in the previous study. We run the MCMC chain for $40,000$ iterations, discard the first half, and retain samples every 10 iterations for each model.

As shown in Figure B.4 and the in-sample ARIs in Table B.1, the STGP model does

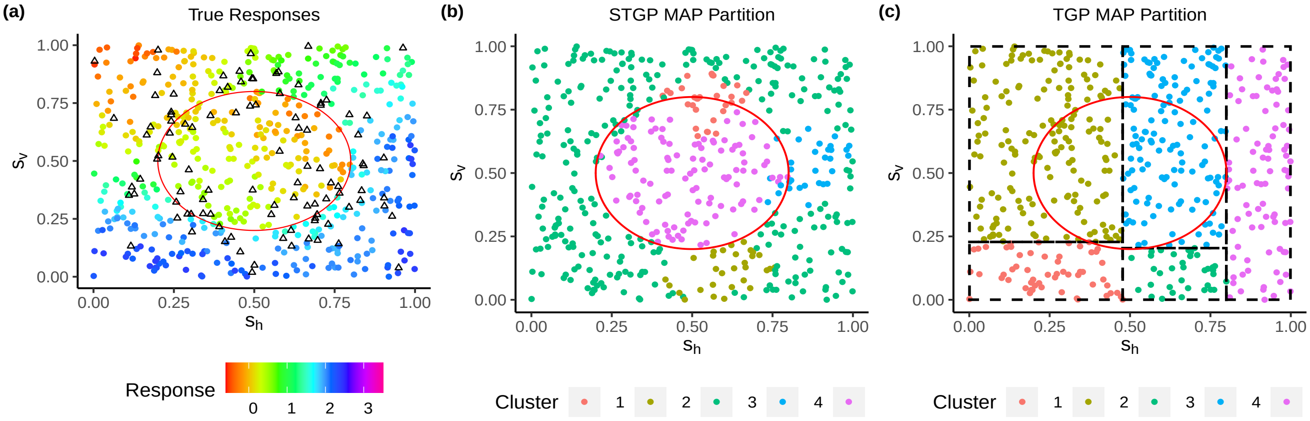

Figure B.4: (a) True $y(\mathbf{s})$ of the training data used in Section B.3.3. (b, c) MAP partition estimates given by STGP and TGP. Dots represents locations in the training data. The true boundary is marked by the red circle. Lines in Panel (c) represent boundaries of the estimated partition.

Table B.1: Performance metrics of STGP and its competitive methods in Simulation Study 2. CRPS_y and LogS_y are averaged over $r = 100$ hold-out locations. Bold numbers indicate the best performance.

	STGP ($L = 1$)	STGP ($L = 3$)	STGP ($L = 5$)	TGP	NSGP	SGP
In-sample ARI	**0.578**	**0.578**	**0.578**	0.022	—	—
Hold-out ARI	**0.322**	0.275	0.233	0.011	—	—
MSPE_y	0.092	**0.048**	0.056	0.057	0.074	0.061
Mean CRPS_y	0.122	**0.089**	0.094	0.116	0.121	0.124
Mean LogS_y	1.237	**-0.576**	-0.553	-0.199	-0.150	0.079

reasonably well in recovering the true partition, while the partition estimate from TGP does not agree with the true one. As for the hold-out locations, the ARIs in the second row of Table B.1 suggest that STGP models overall outperform TGP in terms of predicting cluster memberships.

The out-of-sample prediction performance is summarized in Rows 3-5 of Table B.1. Similarly to the isotropic case, the STGP models with $L = 3$ and 5 demonstrate superior performance in prediction compared with their competitors, evidenced by the lower MSPEs and scores. However, we note the underperformance of STGP with $L = 1$. This is because this model gives several small clusters around the true boundary (see Figure B.4(a)), and hence introduces large errors at some locations near the boundary due to misclassification. Fortunately, for the reasons discussed in Section 3.5, setting $L = 3$ or 5 can considerably mitigates this problem. We therefore recommend proceeding with caution when using $L = 1$.

B.4 Supplementary Results on Real Data Analysis

The prior settings of STGP are the same as in Supplementary Section B.3.3 except that we use a half-Cauchy prior for the spatial ranges, and an $\text{IG}(2, 2)$ prior for $\bar{\sigma}^2$. We work with Lambert conformal conic projection coordinates under which the Euclidean distances approximate the great-circle distances. The spatial graph is again constructed via Delaunay triangulation with edges longer than 5 removed.

As in the simulation studies, we compare the STGP model with TGP and NSGP. We use the same settings as in Supplementary Section B.3.3, except that in NSGP we use 48 regular grids as knots for $\sigma^2(\mathbf{s})$ and assume the mean function has a linear regression form using spatial coordinates as predictors.

The MCMC chains for all models are run for $50,000$ iterations, with the first half as burn-in, and thinned by preserving samples every 5 iterations.

We first examine the predictive surfaces and SDs from all models shown in Figure B.5 at 2649 equally spaced points. The posterior mean predictive surfaces from all three models look similar, except that the surface from the the STGP model with $L = 3$ is smoother than the others. This is probably because TGP uses a less smooth separable exponential covariance function and NSGP assumes a sophisticated GP model on $\sigma^2(\mathbf{s})$. The posterior predictive SD surfaces also share a common pattern: the uncertainty is lower in the central CONUS but much higher in the western region, especially in the southwestern area. The SD of the STGP model near the boundary between MAP Clusters 2 and 3 (see also Figure 3.4(b)) is high, suggesting there may be an abrupt change in the true field across this boundary. Besides, the overall predictive SD within Cluster 2 appears to be larger than that of the other two clusters, possibly due to its smaller cluster size. Note that the uncertainty from

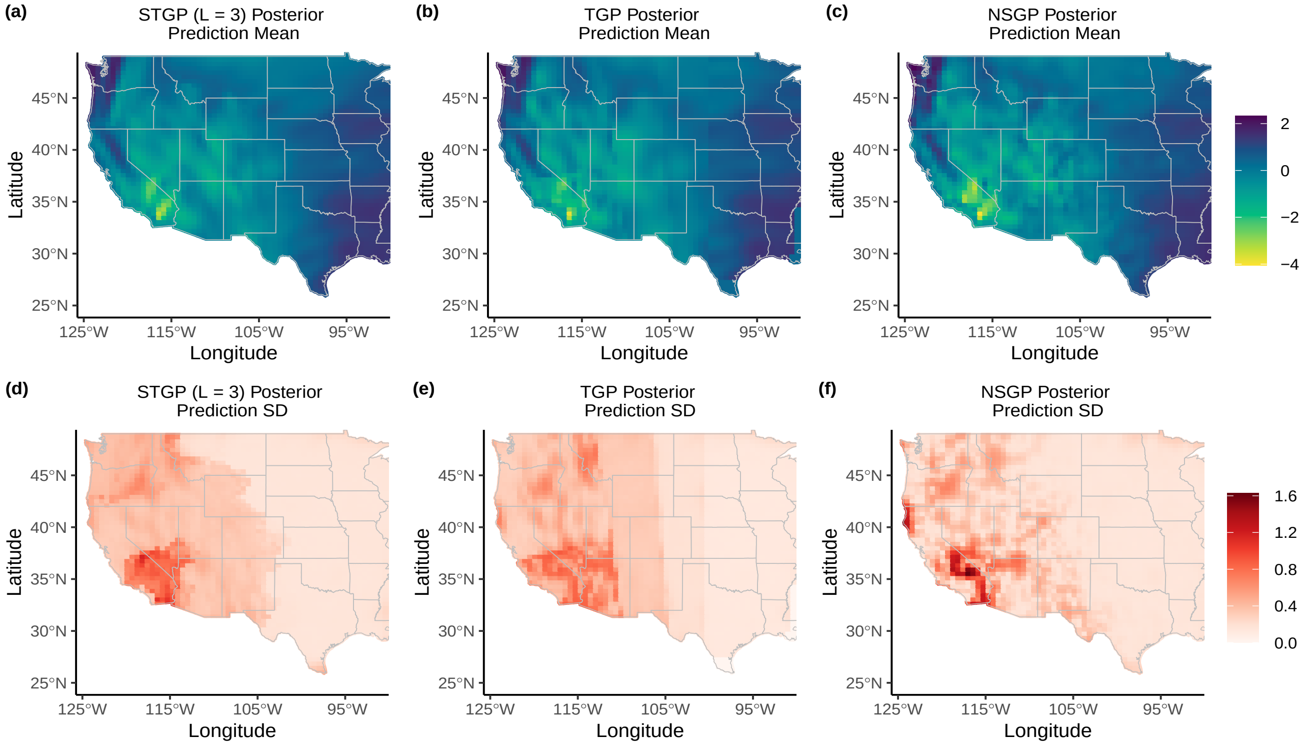

Figure B.5: Maps of (a-c) posterior prediction mean and (d-f) posterior prediction SD of the precipitation data from STGP with $L = 3$, TGP, and NSGP (all in log mm/day).

NSGP at the northwestern coast of California, where few GHCN-D stations are located, is higher than the ones from STGP and TGP.

Next, we provide predictive performance results based on all $r_2 = 175$ hold-out locations that are not near the Rocky Mountains, to examine the prediction performance for points that might be mostly within locally stationary clusters. Figure B.6 visualizes the hold-out locations, and Table B.2 shows the prediction performance metrics based on them. The results suggest that the STGP models have comparable prediction performance with TGP, which is expected as both models assume similar stationary GPs in the interior of each cluster. The results from NSGP are also comparable to the other two methods possibly because the log precipitation rates at these locations are relatively stationary.

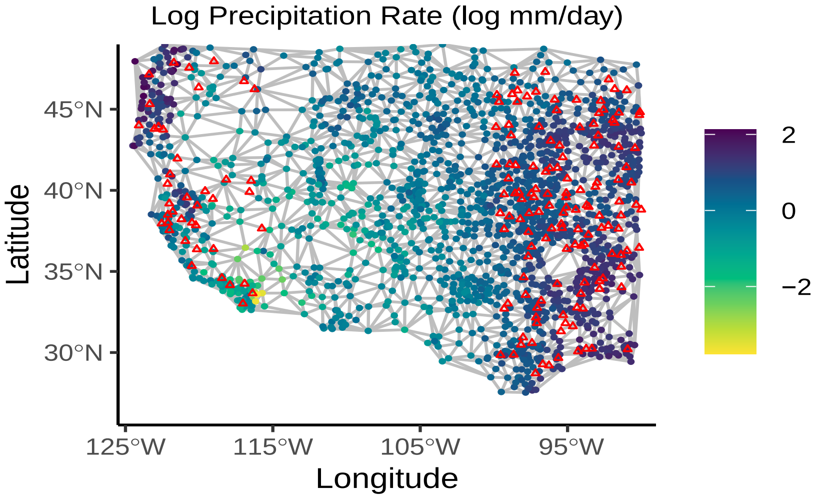

Figure B.6: Log precipitation rate measured at $n = 1689$ GHCN-D stations and the Delaunay triangulation graph used for model fitting. $r_2 = 175$ hold-out locations that are not near the Rocky Mountains are marked as red triangles.

Table B.2: Prediction performance metrics for the precipitation data on $r_2 = 175$ hold-out locations over the CONUS that are not near the Rocky Mountains. CRPS and LogS are averaged over 175 hold-out locations. Bold numbers indicate the best performance.

	STGP $(L = 1)$	STGP $(L = 3)$	STGP $(L = 5)$	TGP	NSGP
MSPE	0.032	0.032	0.032	0.034	**0.031**
Mean CRPS	0.089	0.089	0.089	0.090	**0.086**
Mean LogS	-0.484	-0.485	-0.484	**-0.521**	-0.507

APPENDIX C

SUPPLEMENTARY MATERIALS FOR CHAPTER 4 [*]

This appendix provides supplementary details and results of BAST. Section C.1 contains additional details on Bayesian estimation and prediction. Supplementary simulation details and results including hyperparameter tuning and computation time can be found in Section C.2. Finally, Section C.3 provides the proof of Proposition 4.1.

C.1 Details on Bayesian Inference

C.1.1 Estimation

This appendix provides details on the Markov chain Monte Carlo (MCMC) algorithm discussed in Section 4.3.1. We use $\mathbf{g}_m$ to denote the n-dimensional vector of fitted values at the training locations $\mathcal{S}$ from the mth RST partition, that is, the ith element of $\mathbf{g}_m$ is $g(\mathbf{s}_i|\pi_m, \mathcal{T}_m, k_m, \boldsymbol{\mu}_m)$. Let $\mathbf{X}_{\pi_m}$ be an $n \times k_m$ binary matrix where the (i, j)th element is 1 if and only if $\mathbf{s}_i$ is in the jth cluster under the partition π_m. We write the partial residual term for the mth RST partition as

$$\mathbf{r}_m = \mathbf{Y} - \sum_{\ell \neq m} \mathbf{g}_\ell.$$

Recall that our MCMC algorithm proceeds by successively sampling $(\pi_1, \mathcal{T}_1, k_1, \boldsymbol{\mu}_1)$,. . ., $(\pi_M, \mathcal{T}_M, k_M, \boldsymbol{\mu}_M)$, and σ^2 from their respective full conditional distributions. To sample from $p(\pi_m, \mathcal{T}_m, k_m, \boldsymbol{\mu}_m|-)$ for each $m = 1, \ldots, M$, we first sample the RST partition with $\boldsymbol{\mu}_m$ analytically integrated out, by performing a birth, a death, a change, or a hyper move with probability $r_b(k_m) = 0.3$, $r_d(k_m) = 0.3$, $r_c(k_m) = 0.3$, and $r_h(k_m) = 0.1$, respectively. Adjustments are made to the probabilities for the boundary cases where $k_m = 1$ and $k_m =$

[*] Reprinted from Luo, Z. T., Sang, H., and Mallick, B. (2021a). BAST: Bayesian Additive Regression Spanning Trees for Complex Constrained Domain. *Advances in Neural Information Processing Systems*, 34. The authors hold the copyright.

$\bar{k}$. This probability specification works well in our experiments, but one can modify it if desired. For the first three moves, the Metropolis-Hastings (M-H) acceptance ratio involves the integrated likelihood of $\mathbf{Y}$ given by

$$\mathcal{L}(\mathbf{Y}|\pi_m, \mathcal{T}_m, k_m, -) \propto |\mathbf{P}_{\pi_m}|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{r}_m^\mathsf{T}\mathbf{P}_{\pi_m}^{-1}\mathbf{r}_m\right),$$

where $\mathbf{P}_{\pi_m} = \sigma^2\mathbf{I}_n + \sigma_\mu^2\mathbf{X}_{\pi_m}\mathbf{X}_{\pi_m}^\mathsf{T}$. The Sherman-Woodbury-Morrison formula is applied to simplify the computation of $\mathbf{P}_{\pi_m}^{-1}$ and $|\mathbf{P}_{\pi_m}|^{-1/2}$ as $\mathbf{X}_{\pi_m}\mathbf{X}_{\pi_m}^\mathsf{T}$ has a reduced rank k_m.

Conditional on a sample of $(\pi_m, \mathcal{T}_m, k_m)$, we sample $\boldsymbol{\mu}_m$ from $p(\boldsymbol{\mu}_m|\pi_m, \mathcal{T}_m, k_m, -)$, which is given by

$$[\boldsymbol{\mu}_m|\pi_m, \mathcal{T}_m, k_m, -] \sim \mathrm{N}_{k_m}\left(\mathbf{Q}_m\mathbf{b}_m, \mathbf{Q}_m\right),$$

where $\mathbf{Q}_m = \left(\frac{1}{\sigma^2}\mathbf{X}_{\pi_m}^\mathsf{T}\mathbf{X}_{\pi_m} + \frac{1}{\sigma_\mu^2}\mathbf{I}_{k_m}\right)^{-1}$ and $\mathbf{b}_m = \mathbf{X}_{\pi_m}^\mathsf{T}\mathbf{r}_m/\sigma^2$.

Finally, we sample σ^2 from its inverse-gamma full conditional given by

$$[\sigma^2|-] \sim \mathrm{IG}\left(\frac{n+\nu}{2}, \frac{1}{2}\left[\nu\lambda_s + \|\mathbf{Y} - \sum_{m=1}^{M}\mathbf{g}_m\|^2\right]\right),$$

where $\|\cdot\|$ is the Euclidean norm.

C.1.2 Prediction in Two-dimensional Constrained Domains

In this subsection we provide details on specifying the neighbor set $N_\mathbf{u}$ for prediction at an unobserved location $\mathbf{u}$ in a constrained domain $\mathcal{M} \subset \mathbb{R}^2$. A constrained Delaunay triangulation (CDT) mesh can be constructed on $\mathcal{M}$ such that every unobserved location of interest is contained in a triangle. In the case where at least one triangle vertex is in $\mathcal{S}$, $N_\mathbf{u}$ is specified as those triangle vertices that belong to $\mathcal{S}$. Prediction at $\mathbf{u}$ is then performed as stated in Section 4.3.2.

In the extreme case where no triangle vertex is in $\mathcal{S}$, we choose $N_\mathbf{u}$ to be all the triangle vertices (which lie on the domain boundary). To sample the cluster membership of $\mathbf{u}$, we need to determine the cluster memberships for vertices on the domain boundary, which can be done by, for instance, assigning a boundary vertex to the same cluster as its nearest vertex in $\mathcal{S}$ with respect to the graph distance in the CDT mesh (when the number of vertices in the CDT graph is large, we expect this to well approximate the geodesic distance). Once we obtain the cluster memberships for boundary vertices, we can sample $z_m(\mathbf{u})$ from the cluster memberships of the vertices in $N_{\mathbf{u}}$ as in Section 4.3.2.

C.2 Supplementary Simulation Results

We implement BAST in `R` and fit BART and SFS using `R` packages `BART`[†] (McCulloch et al., 2019) and `mgcv`[‡] (Wood, 2017), respectively. The code for inGP is adopted from `https://github.com/mu2013/Intrinsic-GP-on-complex-constrained-domain`. Experiments are performed on a Linux machine with two Intel Xeon E5-2680 v4 processors and 64GB memory.

C.2.1 U-shape Example

C.2.1.1 Comparison to BART with Larger Numbers of Weak Learners

To demonstrate that BAST is more efficient than its binary treed competitors in recovering irregularly shaped regions where discontinuities happen in complex domains, we compare BAST with $M = 20$ to BART with various numbers of weak learners. The experiment setup is the same as in Section 4.4.1 except for the number of binary decision trees used in BART.

As shown in Table C.1, BAST outperforms BART even when BART uses more weak learners, confirming that BART needs much more rectangular partitions to approximate irregularly shaped discontinuity boundaries, while BAST can recover them with only a few RST edge cuts.

C.2.1.2 Hyperparameter Selection and Sensitivity

We consider selecting hyperparameters of BAST via cross-validation (CV) in the U-shape example with true noise standard deviation $\sigma = 0.1$. More specifically, for each replicate

[†] License: GPL (>= 2)

[‡] License: GPL (>= 2)

Table C.1: Prediction performance of BAST with $M = 20$ weak learners in the U-shape example. Results of BART with various larger numbers of weak learners M are included for comparison. Standard errors are given in parentheses.

		BAST ($M = 20$)	BART ($M = 50$)	BART ($M = 100$)	BART ($M = 200$)
$\sigma = 0.1$	MSPE	**0.189** (0.001)	1.430 (0.049)	1.302 (0.037)	1.219 (0.036)
	MAPE	**0.188** (0.001)	0.408 (0.006)	0.382 (0.005)	0.380 (0.004)
	Mean CRPS	**0.142** (0.001)	0.353 (0.006)	0.324 (0.004)	0.318 (0.003)
$\sigma = 0.5$	MSPE	**0.464** (0.006)	1.694 (0.051)	1.628 (0.039)	1.532 (0.023)
	MAPE	**0.491** (0.004)	0.682 (0.007)	0.695 (0.005)	0.711 (0.005)
	Mean CRPS	**0.371** (0.003)	0.557 (0.006)	0.553 (0.005)	0.554 (0.004)
$\sigma = 1$	MSPE	**1.283** (0.018)	2.546 (0.054)	2.441 (0.035)	2.429 (0.032)
	MAPE	**0.888** (0.007)	1.085 (0.007)	1.099 (0.007)	1.120 (0.007)
	Mean CRPS	**0.693** (0.006)	0.870 (0.007)	0.861 (0.006)	0.870 (0.006)

Table C.2: Candidate values of hyperparameters for CV in the U-shape example.

Method	Hyperparameter	Candidate values
BAST	# of weak learners M	20, 30, 50
	Maximum # of clusters per partition $\bar{k}$	5, 10
	$\boldsymbol{\mu}$-prior shrinkage parameter a	1, 2, 3
BART	# of weak learners M	50, 100, 200
	$\boldsymbol{\mu}$-prior shrinkage parameter a	1, 2, 3

data set, we choose the number of weak learners M, the maximum number of clusters in each RST partition $\bar{k}$, and the shrinkage parameter a that controls prior concentration around zero for $\boldsymbol{\mu}_m$ using 5-fold CV within the training data based on MSPE. The candidate values for each hyperparameter are summarized in Table C.2, and a total of 18 hyperparameter combinations are considered for BAST. For comparision, we also choose the number of weak learners and the prior shrinkage parameter of $\boldsymbol{\mu}_m$ for BART using 5-fold CV, and their candidate values can be also found in Table C.2.

Table C.3 shows the performance of BAST and BART using the hyperparameters chosen by CV (referred to as BAST-cv and BART-cv, respectively). As a benchmark, the performance metrics for BAST and BART using the hyperparameters in Section 4.4.1 are also included (referred to as BAST-default and BART-default, respectively). The fine-tuned

Table C.3: Prediction performance of BAST and BART with and without CV in the U-shape example under noise level $\sigma = 0.1$. Standard errors are given in parentheses.

	BAST-cv	BAST-default	BART-cv	BART-default
MSPE	**0.186** (0.001)	0.189 (0.001)	1.277 (0.043)	1.541 (0.075)
MAPE	**0.182** (0.001)	0.188 (0.001)	0.390 (0.005)	0.436 (0.010)
Mean CRPS	**0.135** (0.002)	0.142 (0.001)	0.331 (0.005)	0.380 (0.009)

BAST-cv achieves better performance than BAST-default as expected, but the performance of them is close to each other, suggesting that BAST is robust to the choices of hyperparameters in this example. Both versions of BAST outperform BART with and without hyperparameter selection.

Next, we further investigate the sensitivity of the performance of BAST to hyperparameters M, $\bar{k}$, and λ_k (the mean parameter of the truncated Poisson prior for k), and how they interact with each other. In general, for large M, one may prefer smaller λ_k and $\bar{k}$ to prevent overfitting and encourage better mixing performance; for small M, one may afford larger λ_k and $\bar{k}$ which may lead to better fitting. Below, we show additional simulation results with different values of M, λ_k, and $\bar{k}$ using the data set in Figure 4.4(b).

Table C.4(a) shows the MSPE for various values of M with a fixed $\lambda_k = 4$ and a fixed $\bar{k} = 10$. The prediction performance of BAST appears to be robust to M except for extremely small M. Increasing M slightly improves the performance until the training data is overfitted. Next, we fix $\lambda_k = 4$ and examine the MSPEs for different combinations of M and $\bar{k}$ shown in Table C.4(b). Again, the performance of BAST does not appear to be sensitive to the choices of M or $\bar{k}$. For a fixed M, increasing $\bar{k}$ improves out-of-sample performance until the model becomes too complex and overfits the training data. As expected, the optimal $\bar{k}$ for larger M is smaller. Finally, we consider varying λ_k while fixing $M = 20$ and $\bar{k} = 10$. As shown in the Table C.4(c), the MSPEs for different values of λ_k are comparable to each other, and the optimal MSPE is achieved with a moderate value $\lambda_k = 4$.

Table C.4: MSPE of BAST under different settings of M, $\bar{k}$, and λ_k in a U-shape domain data set with noise level $\sigma = 0.1$.

(a) MSPE under different values of M

$M = 1$	$M = 5$	$M = 10$	$M = 20$	$M = 30$	$M = 50$
25.54	0.203	0.196	0.192	0.186	0.188

(b) MSPE under different combinations of M and $\bar{k}$

	$\bar{k} = 5$	$\bar{k} = 10$	$\bar{k} = 15$
$M = 20$	0.189	0.192	0.184
$M = 30$	0.188	0.186	0.191
$M = 50$	0.188	0.188	0.190

(c) MSPE under different values of λ_k

$\lambda_k = 2$	$\lambda_k = 4$	$\lambda_k = 6$	$\lambda_k = 8$
0.199	0.192	0.193	0.194

C.2.1.3 Computation Time

Finally, we report in Table C.5 the average computation times (in seconds) of BAST and its competing methods over 50 simulated data sets in Section 4.4.1 with noise level $\sigma = 0.1$. The inference of BAST and BART is based on MCMC, and we remark that BART in the `R` package `bart` is implemented efficiently in `C++` while BAST is implemented in pure `R`. The inference for SFS in the `R` package `mgcv` is based on an efficient optimization algorithm for point estimations only as opposed to a full MCMC inference with uncertainty quantifications, and hence achieves the fastest computation time. The model fitting of inGP requires expensive Brownian motion simulations and thus takes longer time than BAST does.

A more computationally efficient implementation of BAST is under active investigation. Our preliminary `C++` implementation can reduce the computation time from 651.49 seconds to 53.58 seconds. As mentioned in Section 4.3.1, computation can be further improved by fixing spanning trees during MCMC. We refit BAST for the 50 simulated data sets in Section 4.4.1 with noise level $\sigma = 0.1$ by using *different* but *fixed* spanning trees for each weak learner. While the average prediction performance remains comparable (MSPE = 0.190,

Table C.5: Average computation time (in seconds) over 50 simulated data sets in the U-shape example under noise level $\sigma = 0.1$.

BAST (in `R`)	BART	SFS	inGP
651.49 sec.	15.83 sec.	0.68 sec.	787.32 sec.

Table C.6: Prediction performance of BAST and its competing methods in the bitten torus example under different noise levels. Standard errors are given in parentheses.

		BAST	BART	inGP
$\sigma = 0.5$	MSPE	**0.754** (0.008)	1.358 (0.038)	2.601 (0.033)
	MAPE	**0.584** (0.003)	0.682 (0.006)	1.240 (0.010)
	Mean CRPS	**0.405** (0.003)	0.567 (0.006)	—
$\sigma = 1$	MSPE	**1.568** (0.020)	2.378 (0.050)	4.628 (0.445)*
	MAPE	**0.960** (0.007)	1.092 (0.009)	1.648 (0.067)*
	Mean CRPS	**0.706** (0.006)	0.904 (0.009)	—

* The results for inGP under $\sigma = 1$ are based on 49 replicates due to numerical errors in one replicate data set.

MAPE = 0.194, and mean CRPS = 0.145; also see Table 4.1 for baseline performance), the computation time is reduced to 16.82 seconds using the `C++` implementation, which is comparable to BART.

C.2.2 Bitten Torus Example

We consider the bitten torus example in Section 4.4.2 with two additional noise levels $\sigma = 0.5$ and $\sigma = 1$. The results are summarized in Table C.6. Consistent to the findings under the noise level $\sigma = 0.1$, BAST performs the best among all three methods.

As in C.2.1, we also experiment with choosing hyperparameters via 5-fold CV for the data sets with true noise level $\sigma = 0.1$. In addition to the BAST hyperparameters in Table C.2, we also select K, the size of the predictive neighbor set $N_{\mathbf{u}}$ discussed in Section 4.3.2, from its candidate values $\{3, 4, 5, 6\}$. As shown in Table C.7, BAST outperforms BART in both CV and default settings. Our results again confirm that BAST performs reasonably well even without hyperparameter tuning.

Table C.7: Prediction performance of BAST and BART with and without CV in the bitten torus example under noise level $\sigma = 0.1$. Standard errors are given in parentheses.

	BAST-cv	BAST-default	BART-cv	BART-default
MSPE	**0.463** (0.008)	0.487 (0.002)	0.850 (0.020)	1.115 (0.041)
MAPE	**0.287** (0.004)	0.307 (0.001)	0.370 (0.004)	0.406 (0.009)
Mean CRPS	**0.216** (0.003)	0.225 (0.002)	0.310 (0.004)	0.355 (0.008)

C.3 Proof of Proposition 4.1

Proof of Proposition 4.1. For any spatially continuous partition $\pi(\mathcal{S})$ with k clusters, it follows from Proposition 2 of Luo et al. (2021b) that there exists a spanning tree $\mathcal{T}$ of $\mathcal{G}$ and a set of $k-1$ edges in $\mathcal{T}$ that induce $\pi(\mathcal{S})$. Hence, conditional on $\mathcal{T}$, the conditional probability for $\pi(\mathcal{S})$ is strictly positive due to (4.2) and (4.4). To show $\mathcal{T}$ is within the support of (4.3), note that $\mathcal{T}$ is the MST of $\mathcal{G}$ given the edge weights satisfying $\omega_e \in (0, 1/2)$ if $e \in \mathcal{E}_{\mathcal{T}}$ and $\omega_e \in (1/2, 1)$ if $e \notin \mathcal{E}_{\mathcal{T}}$. This completes the proof. □

APPENDIX D

SUPPLEMENTARY MATERIALS FOR CHAPTER 5

D.1 Details on Spanning Tree Bipartitions

As discussed in Section 5.2.1, the spanning tree graph $\mathcal{G}_T^*$ on reference knots $\mathcal{S}^*$ can be obtained by finding the geodesic distance based MST of a graph $\mathcal{G}^* = (\mathcal{S}^*, \mathcal{E}_0^*)$, which is constructed following Luo et al. (2021a). In practice, however, when the number of knots is small or when the shape of $\mathcal{M}$ is highly irregular, the methods in Luo et al. (2021a) may result in a disconnected $\mathcal{G}^*$. To overcome this, one can augment $\mathcal{E}_0^*$ to make $\mathcal{G}^*$ connected using Algorithm 3.

```
Algorithm 3: Connecting connected components in $\mathcal{G}^*$
    Input: a graph $\mathcal{G}^* = (\mathcal{S}^*, \mathcal{E}_0^*)$ with $N_c$ connected components.
    Initialize $\mathcal{C}$ to be the vertices in one connected component of $\mathcal{G}^*$.
    for $i = 1$ to $N_c - 1$ do
        Find the pair of vertices $\mathbf{v}_1 \in \mathcal{C}$ and $\mathbf{v}_2 \in \mathcal{S}^* \setminus \mathcal{C}$ that has the minimal geodesic
        distance.
        Add the edge $(\mathbf{v}_1, \mathbf{v}_2)$ to $\mathcal{E}_0^*$.
        Set $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$, where $\mathcal{C}'$ is the connected component containing $\mathbf{v}_2$.
    end for
    Output: a connected graph $\mathcal{G}^*$.
```

The constructions of the aforementioned graphs and $\pi_2(\mathcal{M}_\eta)$ rely on the geodesic distance d_g in $\mathcal{M}$. For many manifolds, d_g has no analytical form. Fortunately, we can approximate d_g between any two locations in a way similar to Isomap algorithm (Tenenbaum et al., 2000). To be more specific, we construct a dense weighted nearest neighbor graph based on Euclidean distance on some fine grids in $\mathcal{M}$ and the locations of interest, and then approximate the geodesic distance between the two locations by the length of the shortest path between them in the dense graph.

D.2 Details on Bayesian Inference

This appendix provides details on the Markov chain Monte Carlo (MCMC) algorithm in Section 5.3. Given data $(\mathbf{s}_1, \mathbf{x}_1, Y_1), \cdots, (\mathbf{s}_n, \mathbf{x}_n, Y_n)$, let $\mathbf{g}_m$ be the vector of in-sample fitted values from the mth weak learner, i.e., the ith element of $\mathbf{g}_m$ is $g(\mathbf{s}_i, \mathbf{x}_i | T_m, \boldsymbol{\mu}_m)$. Define the partial residual from the mth weak learner as

$$\mathbf{r}_m = \mathbf{Y} - \sum_{k \neq m} \mathbf{g}_k.$$

As discussed in Section 5.3, our MCMC sampler successively draw samples from the full conditional distributions of $(T_1, \boldsymbol{\mu}_1), \ldots, (T_M, \boldsymbol{\mu}_M)$, and σ^2. To sample from each $p(T_m, \boldsymbol{\mu}_m | -)$, we proceed in two steps. First, we update T_m using a Metropolis-Hastings (MH) sampler by drawing T_m from $p(T_m | -)$, the full conditional distribution of T_m with $\boldsymbol{\mu}_m$ integrated out. Specifically, we propose a new psMDT T_m^* by a growing or a pruning move as detailed in Section 5.3. In a *growing* move, letting η be the node we split, the MH acceptance probability is given by

$$\min\left\{1, \; \frac{\alpha(1+d_\eta)^{-\beta}[1-\alpha(2+d_\eta)^{-\beta}]^2}{1-\alpha(1+d_\eta)^{-\beta}} \cdot \frac{N_s}{N_m} \cdot \frac{\mathcal{L}(\mathbf{Y}|T_m^*, -)}{\mathcal{L}(\mathbf{Y}|T_m, -)}\right\}, \tag{D.1}$$

where N_s is the number of terminal nodes in T_m, N_s is the number of non-terminal nodes with two terminal children in T_m, and $\mathcal{L}(\mathbf{Y}|T_m, -)$ is the likelihood of $\mathbf{Y}$ with $\boldsymbol{\mu}_m$ marginalized out. Thanks to the conjugate prior on $\boldsymbol{\mu}_m$, $\mathcal{L}(\mathbf{Y}|T_m, -)$ can be explicitly evaluated by

$$\mathcal{L}(\mathbf{Y}|T_m, -) \propto |\mathbf{P}_m|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{r}_m^\top \mathbf{P}_m^{-1} \mathbf{r}_m\right),$$

where $\mathbf{P}_m = \sigma^2 \mathbf{I}_n + \sigma_\mu^2 \mathbf{Z}_m \mathbf{Z}_m^\top$ and $\mathbf{Z}_m$ is an $n \times \ell_m$ binary matrix whose (i,j)th element is 1 if and only if the ith observation is assigned to the jth terminal node of T_m. In practice, utilizing the fact that $\mathbf{Z}_m$ has reduced rank ℓ_m, we use Sherman-Woodbury-Morrison formula to simplify the computation of $|\mathbf{P}_m|$ and $\mathbf{P}_m^{-1}$. The MH acceptance probability of a pruning move is analogous to (D.1).

The second step to sample from $p(T_m, \boldsymbol{\mu}_m|-)$ is to draw $\boldsymbol{\mu}_m$ from $p(\boldsymbol{\mu}_m|T_m, -)$, which admits a closed form

$$[\boldsymbol{\mu}_m|T_m, -] \sim \mathrm{N}_{\ell_m}\left(\mathbf{Q}_m^{-1}\mathbf{b}_m, \mathbf{Q}_m^{-1}\right),$$

with $\mathbf{Q}_m = \mathbf{Z}_m^\mathsf{T}\mathbf{Z}_m/\sigma^2 + \mathbf{I}_{\ell_m}/\sigma_\mu^2$ and $\mathbf{b}_m = \mathbf{Z}_m^\mathsf{T}\mathbf{r}_m/\sigma^2$.

Finally, the full conditional of σ^2 is an inverse gamma distribution of the form

$$[\sigma^2|-] \sim \mathrm{IG}\left(\frac{n+\nu}{2}, \frac{1}{2}\left[\nu\lambda_s + \|\mathbf{Y} - \sum_{m=1}^{M}\mathbf{g}_m\|^2\right]\right),$$

where $\|\cdot\|$ is the Euclidean norm.

D.3 Supplementary Simulation Details

D.3.1 Details on Simulation Setup

We consider a two-dimensional U-shape domain $\mathcal{M}$ and generate uniform random locations $\mathbf{s} = (s_h, s_v)$ in $\mathcal{M}$. Below, we discuss the generation of unstructured features $\mathbf{x}$. In many applications, there is oftentimes spatial dependence among locations within an unstructured feature. To simulate spatially correlated features, we first find a homomorphism $(u_1, u_2) = h(s_h, s_v)$ from $\mathcal{M}$ to a rectangular region in $\mathbb{R}^2$. Then we simulate independent realizations $\{\zeta_1\}, \ldots, \{\zeta_p\}$ from a Gaussian process using Euclidean distance on (u_1, u_2). We further use the transformation $x_j = \Phi(\zeta_j)$ to generate unstructured features within $[0, 1]$, where Φ is the cumulative distribution function of standard Gaussian distribution.

Motivated by Ramsay (2002), we construct a true function as $f(\mathbf{s}, \mathbf{x}) = b_0 + b_1(u_1x_1 + u_2^2)$ for some constants b_0 and b_1, which only depends on the structured features $\mathbf{s}$ and one of the unstructured features. We allow b_0 and b_1 to take different values in different subregions of $\mathcal{M}$ to create discontinuities. Specifically, we divide $\mathcal{M}$ into three subsets separated by a circle:

$$
\begin{aligned}
\mathcal{M}_1 &= \{(s_h, s_v) \in \mathcal{M} : s_h^2 + s_v^2 > 0.9^2 \text{ and } s_h < s_v\},\\
\mathcal{M}_2 &= \{(s_h, s_v) \in \mathcal{M} : s_h^2 + s_v^2 > 0.9^2 \text{ and } s_h > s_v\},\\
\mathcal{M}_3 &= \{(s_h, s_v) \in \mathcal{M} : s_h^2 + s_v^2 \leq 0.9^2\}.
\end{aligned}
$$

We set $b_0 = -4$ and $b_1 = 1$ in $\mathcal{M}_1$, $b_0 = 4$ and $b_1 = 1$ in $\mathcal{M}_2$, and $b_0 = 0$ and $b_1 = -0.5$ in $\mathcal{M}_3$.

D.3.2 Supplementary Results

In this appendix, we compare the predictive uncertainty using the same simulated data used in Section 5.4.1 under the setting of $p = 2$ and $\sigma = 0.1$. The predictive uncertainty at different spatial locations is shown in Figure D.1. As expected, the posterior predictive standard deviation (SD) from BAMDT is higher around the discontinuity surfaces, reflecting the uncertainty due to the unknown discontinuities. The uncertainty measures from BART and GP regression, however, fail to capture this. In the predictive SD for BART, one can observe some artificial axis-parallel high uncertainty regions probably resulting from univariate splits on $\mathbf{s}$. The uncertainty of GP regression at unobserved locations is generally higher, possibly due to mis-specification of the model especially in the mean function.

We also examine the sensitivity of BAMDT's prediction performance to the hyperparameters using a data set under the setting of $p = 2$ and $\sigma = 0.1$. We consider different values of the number of weak learners M, the number of reference knots t, and the prior probability for performing a multivariate split p_m. Prediction metrics are shown in Table D.1. Overall, MAPE and mean CRPS are generally robust to different hyperparameter settings. There is some variability in MSPE, probably due to the relatively large prediction errors near the discontinuity surfaces. When the performance near discontinuity boundaries is a concern, we recommend using standard hyperparameter selection methods such as cross-validation to fine tune the model.

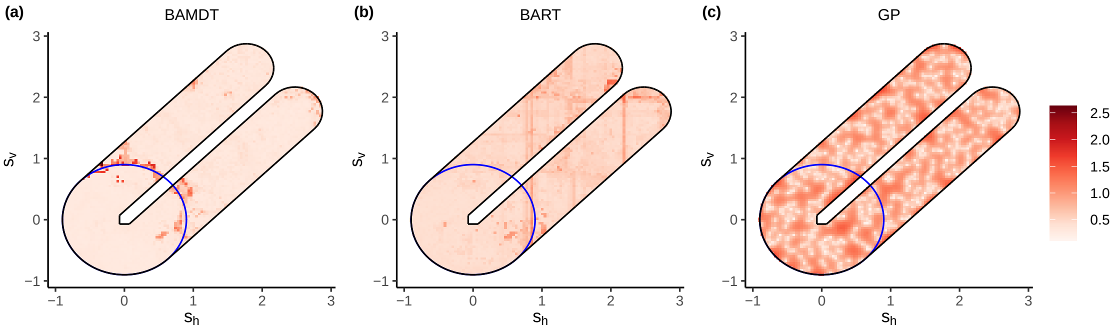

Figure D.1: Posterior predictive standard deviation of (a) BAMDT, (b) BART, and (c) GP regression in the setting of $p = 2$ and $\sigma = 0.1$. Blue circles indicate discontinuity surfaces in the true function projected to $\mathcal{M}$.

D.4 Supplementary Real Data Analysis

In this appendix, we provide more analysis on the Sacramento housing price data.

We first examine feature importance in BAMDT. 44.92% of the splits are attributed to the structured feature $\mathbf{s}$, suggesting that a large part of the variation in Sacramento housing price can be explained by the spatial component of the model. The square footage feature is the second important feature in BAMDT, followed by the number of bathrooms and bedrooms. A similar feature importance pattern is found using BART.

In Section 5.4.2, we have examined the marginal effect of the spatial locations $\mathbf{s}$. Below, we focus on the marginal effect of square footage. We choose five representative locations in Downtown Sacramento (green), North Natomas (cyan), North Sacramento (red), Valley Hi / North Laguna (blue), and Elk Grove (pink), as shown in Figure D.2(a). Figure D.2(b) shows the predicted price of houses with three bedrooms, two bathrooms, and various square footage. As expected, there is a positive nonlinear relationship between price and square footage at each location, and there is a noticeable change in the relationship near 1600 square feet. The marginal effect of footage also depends on the locations; Downtown Sacramento has the highest price per square feet, while North Sacramento has the lowest. 95% predictive credible intervals of these two locations are also shown. The credible intervals are wider for

Table D.1: Prediction performance of BAMDT under different settings of hyperparameters.

M	t	p_m	MSPE	MAPE	Mean CRPS
50	100	0.25	0.318	0.268	0.214
100	100	0.25	0.258	0.232	0.178
50	200	0.25	0.475	0.292	0.232
100	200	0.25	0.442	0.288	0.222
50	100	0.50	0.222	0.236	0.175
100	100	0.50	0.305	0.239	0.183
50	200	0.50	0.395	0.300	0.244
100	200	0.50	0.409	0.270	0.210
50	100	0.75	0.482	0.305	0.239
100	100	0.75	0.327	0.261	0.199
50	200	0.75	0.457	0.304	0.235
100	200	0.75	0.361	0.259	0.196

larger houses, probably because of the log-transformation of price in the model.

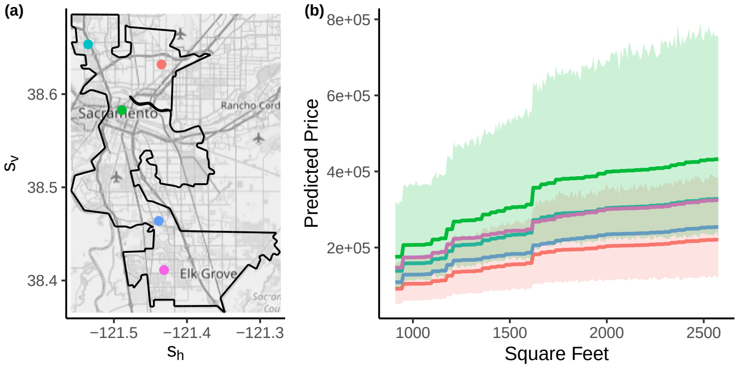

Figure D.2: (a) Map of five representative locations. (b) Predicted price versus square footage of the houses. Colored ribbons represent 95% predictive credible intervals of two representative locations.