Journal of the Royal Statistical Society Series C: Applied Statistics, 2025, **00**, 1–20
https://doi.org/10.1093/jrsssc/qlaf024

ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS
Journal of the Statistics Society
Series C
Applied Statistics

Original Article

Location smoothed Bayesian additive regression trees: a method for interpretable and robust quality assurance of organ contours in radiotherapy treatment planning

Zachary T. Wooten[1], Mary Pham[2], Laurence E. Court[2] and Christine B. Peterson[3]

[1]Department of Statistics, Rice University, 6100 Main St., Houston, TX 77005, USA
[2]Department of Radiation Physics, The University of Texas MD Anderson Cancer Center, 1400 Pressler St., Houston, TX 77030, USA
[3]Department of Biostatistics, The University of Texas MD Anderson Cancer Center, 1400 Pressler St., Houston, TX 77030, USA

Address for correspondence: Zachary T. Wooten, Department of Statistics, Rice University, 6100 Main St., Houston, TX 77005, USA. Email: zachary.t.wooten@rice.edu

Abstract

Deep learning techniques for image segmentation are increasingly used in automating anatomical structure delineation in medical images for radiation treatment planning. Given the critical role these contours play in guiding radiotherapy, it is crucial to flag errors before planning, necessitating robust quality assurance methods for the clinical adoption of automated contours. To address this challenge, we introduce location smoothed Bayesian additive regression trees (lsBART), a novel Bayesian tree-based model for nonparametric scalar on function regression. Our proposed method can identify both relevant functions and important regions within those functions, enabling interpretable, and sparse solutions. We benchmark lsBART on a simulated regression setting with multiple functional predictors, where it achieves a lower root mean squared error than existing alternative methods. In our real data application to identifying errors in kidney contours, we attained a cross-validated area under the curve of 0.905 for detecting unacceptable contours. Using Shapley values, we provide guidance on aspects of the contour in specific regions that led to the contour being flagged, indicating our method's potential clinical utility.

Keywords: Bayesian additive regression trees, functional data analysis, medical imaging, radiotherapy planning, Shapley value

1 Introduction

Radiation therapy treatment planning relies on accurate segmentation of anatomical structures from medical images to ensure precise targeting of radiation to the tumour and to minimize damage to nearby organs. With the emergence of deep learning models such as convolutional neural networks, there has been significant progress in automating the contouring of organs in medical images (Cardenas et al., 2018). This automation has several potential benefits, including reducing the workload of clinicians, minimizing human error, and enhancing healthcare accessibility for low-income communities (Court et al., 2023). However, deep learning models can fail when confronted with unexpected differences in patient anatomy or imaging protocols different from those in their training data. As such, there is a growing need to ensure the reliability of deep learning model outputs. To illustrate this challenge, Figure 1a showcases an axial slice from a cervix

Received: December 1, 2023. **Revised:** February 14, 2025. **Accepted:** March 7, 2025

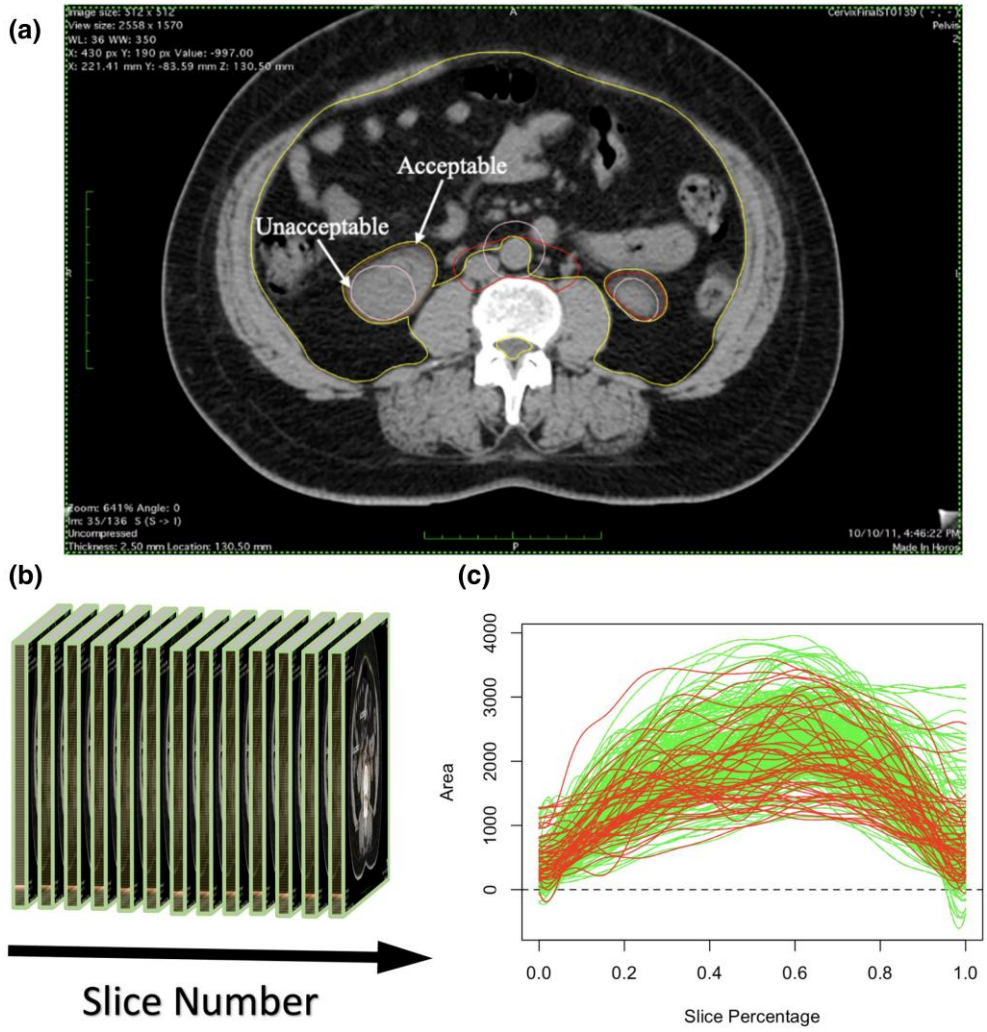

Figure 1. (a) Radiotherapy treatment plan with clinically acceptable and unacceptable kidney contours (Wooten et al., 2023); (b) stacking of axial slices; we take the scalar shape statistic values from each axial slice of the treatment plan and turn the collection of values into a shape statistic functional predictor on $s \in [0, 1]$ representing the percentage of the plan; (c) area functional predictors, where green are extracted from acceptable contours and red from unacceptable contours.

radiotherapy treatment plan. The highlighted arrows differentiate between a clinically acceptable contour of the kidney and an unacceptable contour that misses part of the organ. Since radiation treatment plans are optimized to minimize radiation to critical structures, erroneous organ contours can put patients at increased risk of radiation-associated toxicities.

While current methodologies call for manual inspection of automatically generated contours, a streamlined, automated review process that identifies and highlights problematic contours remains a more desirable solution. Reviewing contours can be particularly time-consuming given that each plan consists of dozens of image slices, corresponding to the three-dimensional organ when stacked (Figure 1b). Various approaches have been proposed for performing contour quality assurance on both human and computer-generated contours (Chen et al., 2015; McIntosh et al., 2013; Rhee et al., 2019; Wooten et al., 2023).

However, these methods have key limitations that hinder their clinical translation: they either depend heavily on the image intensity values, which limits generalization across imaging modalities and institutions; rely on the same modelling approach for the contouring and quality

assurance tasks, which can result in repetitive mistakes between planning and verification stages; or generically flag an entire treatment plan without pinpointing problematic areas for further review. Clinically, an organ contour could be considered unacceptable due to inaccuracies in any image slice. An optimal tool would be robust to differences in imaging platform and directly indicate the erroneous slices, streamlining the review process for clinicians and ensuring precise correction.

To address this challenge, we propose location smoothed Bayesian additive regression trees (lsBART), a novel statistical learning method based on the BART framework (Chipman et al., 2010), which has key advantages over existing approaches for our motivating application. Location smoothed BART offers the flexibility of tree-based models, which can capture nonlinear and interaction effects (Breiman, 2001). Additionally, lsBART promotes sparsity in both functional predictors and specific locations within these predictors, making it ideal for high-dimensional datasets with spatial correlation. To handle the spatial structure in our data, lsBART incorporates smoothing on the probability of feature selection across neighbouring locations, to reflect that the organ contours vary smoothly across neighbouring image slices. To the best of our knowledge, we are the first to apply a new and highly efficient method for computing Shapley values to the Bayesian tree ensemble setting, enabling insight for our real data application on the location and type of potential contouring errors for a specific patient.

To illustrate our proposed method, we apply lsBART in a simulated regression setting and a real data application to kidney contours for use in cervix radiotherapy treatment plans. Importantly, lsBART cannot only identify erroneous contours but also indicate locations that caused the contour to be flagged. Current quality assurance techniques lack the ability to inform radiation treatment planners which slices may contain errors. Furthermore, our method for quality assurance relies solely on shape statistics, making it robust to potential differences in image intensity across institutions and imaging platforms, such as MRI vs. CT. Even within a single modality such as CT, differences in imaging protocols across institutions (such as the use of different tube voltages) may result in different image intensity values.

A central argument of our paper is that the collection of two-dimensional shape statistics from each image slice can be conceptualized as a functional data object, enabling the application of functional data analysis (FDA) tools. Figure 1b illustrates the sequential stacking of axial slices. Due to differences in the slice width and patient size, each observation contains a differing number of slices. When computing a shape statistic, like area, for each axial slice, we derive a vector whose length corresponds to the number of slices encompassing the organ. Through the functional data framework, we are able to leverage the spatial smoothness in our input data to place the observed data on a common grid. Figure 1c presents a series of smooth functional data observations, with 0% marking the organ's lower boundary and 100% denoting its upper boundary.

The rest of the paper is organized as follows. Section 2 provides background on methods for FDA and BART since our framework builds on these methods. Section 3 describes a novel location smoothed BART model. In Section 4, we illustrate the utility of this model on both simulated data and our motivating task of quality assurance for radiotherapy treatment planning contours. We conclude with a discussion in Section 5.

2 Background

2.1 Functional data

Since each patient has a different number of axial image slices, we adopt a FDA framework to leverage smoothness across neighbouring slices and place the observed data onto a common grid. Functional data analysis addresses data characterized by continuous variation across domains such as time. Such data, termed *functional data*, can be represented by smooth functions spanning a designated domain. A dataset with n observations of a functional predictor $X(s)$ and scalar response y can be represented as $(\mathbf{X}(s), \mathbf{y})$, where $s \in S$ represents a range of underlying points. The predictor function for the ith subject $X_i(s)$ can be expressed as a weighted sum of basis functions $X_i(s) \approx \sum_{b=1}^{B} c_{ib}\phi_b(s)$, where c_{ib} is the coefficient for basis function $\phi_b(s)$, for $b = \{1, \ldots, B\}$. The basis functions may be a set of standard functions, such as polynomials or splines, or may be estimated from the data (Ramsay & Silverman, 2005).

2.2 Bayesian additive regression trees

Before introducing lsBART, we briefly review the BART framework. The BART model is a Bayesian machine learning technique adept at capturing intricate relationships between predictors and a response (Chipman et al., 2010). It constructs decision trees within a Bayesian framework, allowing the inclusion of prior information on parameters and enhancing model interpretability. As in classical Bayesian inference, the BART model is defined by a likelihood and set of priors, and inference is performed by sampling from the posterior. Let y_i be a scalar response, $\boldsymbol{x}_i$ be a vector of p covariates, and μ_0 be a known constant which centres y. The default value of μ_0 is typically set to be the mean of y. The BART model assumes $y_i = \mu_0 + f(\boldsymbol{x}_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $f(\boldsymbol{x}_i)$ is modelled as a sum of H decision trees $\sum_{h=1}^{H} g(\boldsymbol{x}_i, \mathcal{T}_h, \mathcal{M}_h)$. Here, $\mathcal{T}_h$ represents a binary tree containing split rules, e.g. $x_i < 0.5$, and $\mathcal{M}_h$ denotes the mean values associated with each leaf node, i.e. for L leaf nodes $\mathcal{M}_h = \{\mu_{h1}, \mu_{h2}, \ldots, \mu_{hL}\}$. Then $g(\boldsymbol{x}_i, \mathcal{T}_h, \mathcal{M}_h) = \mu_{hl}$ represents the step function that maps input vector $\boldsymbol{x}_i$ to a leaf node value μ_{hl} for $l = \{1, \ldots, L\}$ for the hth tree $\mathcal{T}_h$ with leaf parameters $\mathcal{M}_h$.

Various extensions to BART have been proposed that have a bearing on this research. To better handle settings with a large number of irrelevant predictors, Linero (2018) developed the Dirichlet additive regression trees (DART) model, which introduces a sparsity-inducing Dirichlet prior. To handle grouped predictors such as genes within pathways, Du and Linero (2019) proposed overlapping group BART to enforce sparsity across and within groups. Finally, BART with target smoothing is tailored for function-on-scalar regression scenarios (Starling et al., 2020). However, none of these directly address our problem of interest; therefore, we develop BART prediction methodology for multiple functional input variables.

3 Location smoothed Bayesian additive regression trees

3.1 Proposed model

We now describe the lsBART model, which is designed to both select relevant predictor functions and identify regions of interest within those functions for regression and classification. Location smoothed Bayesian additive regression trees extends the BART framework to address the challenges of our motivating application by incorporating a novel prior distribution and smoothing technique that leverages the unique structure of functional data to achieve improved prediction and model inference.

We first introduce the structure of our input data. We observe p predictor functions $X_j(s)$, where $j = \{1, \ldots, p\}$, for each subject. We assume that the function values are available on a common spatial grid with S discrete values, $s = \{s_1, s_2, \ldots, s_S\}$, which can be achieved through the use of a functional basis representation. Then for each functional predictor $X_j(s)$, we have a vector of S discrete values $(X_j(s_1), X_j(s_2), \ldots, X_j(s_S))$. We concatenate the functions for each subject to obtain a $p * S$ vector $\hat{\boldsymbol{x}}_i$, where the *hat* symbol indicates the estimated values obtained from the functional basis representation. For large p or S, achieving model sparsity is crucial and improves interpretability by highlighting the most relevant functional predictors and regions. We can write our input data matrix as

$$\hat{\boldsymbol{X}}_{n\times(p*S)} = \begin{pmatrix} X_1(s_1)_1 & \cdots & X_1(s_S)_1 & X_2(s_1)_1 & \cdots & X_2(s_S)_1 & \cdots & X_p(s_1)_1 & \cdots & X_p(s_S)_1 \\ X_1(s_1)_2 & \cdots & X_1(s_S)_2 & X_2(s_1)_2 & \cdots & X_2(s_S)_2 & \cdots & X_p(s_1)_2 & \cdots & X_p(s_S)_2 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ X_1(s_1)_n & \cdots & X_1(s_S)_n & X_2(s_1)_n & \cdots & X_2(s_S)_n & \cdots & X_p(s_1)_n & \cdots & X_p(s_S)_n \end{pmatrix}.$$

For a single subject, the lsBART model assumes $y_i = \mu_0 + f(\hat{\boldsymbol{x}}_i) + \varepsilon_i$, where y_i is a scalar response variable, $\hat{\boldsymbol{x}}_i$ is the collection of S discrete values from each of the p functional predictors defined above, μ_0 is the known constant which centres y, and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ is Gaussian error. The idea is to estimate f as a sum of decision trees with parameters $(\mathcal{T}_h, \mathcal{M}_h)$, such that $f(\hat{\boldsymbol{x}}_i) \approx g(\boldsymbol{x}_i, \mathcal{T}_1, \mathcal{M}_1) + \cdots + g(\boldsymbol{x}_i, \mathcal{T}_H, \mathcal{M}_H)$. In our approach, we focus on discrete points within the function, rather than the shape of the entire function, while still allowing for spatial correlation. This decision is motivated by several considerations. Namely, by focusing on

discrete points, we can pinpoint specific locations within the functional data that are most relevant to the prediction task. These specific locations allow for enhanced interpretability of the model's output.

3.2 Prior formulation

Now that we have introduced the lsBART sum-of-trees structure, we describe the priors on the model parameters. Specifically, we discuss the priors on the parameters that influence the probability of node splitting, the selection of the function in the splitting rule, the chosen location, the leaf node's final value, and the standard deviation of error. Our proposed lsBART model employs Dirichlet priors to select relevant functional predictors and pinpoint important regions within them. lsBART uses a regularization prior similar to that of the original BART model to regulate the size and fit of $(\mathcal{T}_h, \mathcal{M}_h)$. This ensures that each tree contributes modestly to the overall fit, preventing any single tree from being overly dominant. The novel aspect of our approach lies in our sequential approach to encourage sparsity in both the selection of functions and critical regions. First, we adopt a Dirichlet prior to determine which function to use in the splitting rule from the p functional predictors. Second, we incorporate another Dirichlet prior to select values from the underlying spatial grid, $s = \{s_1, s_2, \ldots, s_S\}$, that are crucial for the prediction task, subject to the constraint that location probabilities are smooth across space. This dual-layered approach induces sparsity in identifying functional predictors and in pinpointing the locations within selected functions, highlighting significant functions and their essential locations probabilities.

3.2.1 Priors for each tree $\mathcal{T}_h$

The tree depth prior (item (1) below) controls tree complexity and prevents overfitting by reducing the likelihood of splits at large depths. The functional predictor selection prior (item (2)) encourages sparsity by favouring a small subset of relevant functional predictors for splitting. Additionally, for each selected predictor, the location selection prior (item (3)) further promotes sparsity across regions. Finally, a cut-off selection prior (item (4)) allows for flexibility in defining decision rules for splitting features, ensuring that the model can capture a wide range of potential split points. In more detail:

1. As the prior on the tree structure, we assume $P(\text{split at node depth } d) = \alpha(1+d)^{-\beta}$ for $\alpha \in (0, 1)$, $\beta \in [0, \infty)$. We use the parameter setting of $\alpha = 0.95$ and $\beta = 2$ as in Chipman et al. (2010). This choice of hyperparameters favours small trees with two to three leaf nodes, regularizing the complexity of the individual trees within the ensemble.
2. Next, we specify the prior on the probabilities that the tree will split based on a certain predictor. Let $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_p\}$ be the vector of inclusion probabilities for each functional predictor. We assume a Dirichlet prior $\boldsymbol{\pi} \sim \text{Dirichlet}(\alpha_{\pi_1}, \alpha_{\pi_2}, \ldots, \alpha_{\pi_p})$, where each α_{π_j} represents the hyperparameter associated with the inclusion probability π_j. As described in Linero (2018), a Dirichlet splitting rule prior encourages sparsity and allows for fully Bayesian inference of feature importance. We set the prior parameters to be uniform across the candidate functions, with $\alpha_{\pi_j} = \frac{1}{p}$.
3. We now describe the prior that controls the selection of locations within each function. Conditional on the selection of the jth predictor function, let the probability of using a specific location in the splitting rule be $\boldsymbol{\tau}_j = \{\tau_{j1}, \tau_{j2}, \ldots, \tau_{jS}\}$. Then $\boldsymbol{\tau}_j \sim \text{Dirichlet}(\alpha_{\tau_{j1}}, \alpha_{\tau_{j2}}, \ldots, \alpha_{\tau_{jS}})$, where each $\alpha_{\tau_{jk}}$ represents the hyperparameter associated with the inclusion probability τ_{jk}. In our prior formulation, we set the prior parameters to be uniform across the candidate locations to ensure that each location has an equal likelihood of being selected, with $\alpha_{\tau_{jk}} = \frac{1}{S}$ for $k = \{1, \ldots, S\}$. The Dirichlet splitting rule encourages sparsity across spatial locations, promoting a focus on specific, relevant regions. We impose a prior constraint that the location probabilities should be smooth over space. We describe our approach to smoothing the $\boldsymbol{\tau}_j$ values in Section 3.3.
4. Lastly, given the selected function $X_j(s)$ and location within the function s_k, we need to specify the threshold used to define the decision rule used for splitting the observations. Since we

assume the function values are continuous, this decision rule will be of the form $x \leq \lambda$ vs. $x > \lambda$. We place a uniform prior on the possible splitting values as in Chipman et al. (2010): $\lambda_{jk} \sim \text{Uniform}(\text{range}(\hat{\mathbf{X}}_{jk}))$, where $\hat{\mathbf{X}}_{jk} = (X_j(s_k)_1, X_j(s_k)_2, \ldots, X_j(s_k)_n)$.

For completeness, we now describe the prior distributions on the leaf node values and the standard deviation of error, where we follow the settings recommended by Chipman et al. (2010).

3.2.2 *Prior for each leaf value* μ_{hl}

Let a single leaf value for a given tree $\mathcal{T}_h$ be denoted as μ_{hl}. Suppose $y_i \in [y_{\min}, y_{\max}]$ for all i and denote the set $\{\mu_{1(i)}, \ldots, \mu_{H(i)}\}$ as the leaf values from each tree corresponding to $\boldsymbol{x}_i$. As described in Sparapani et al. (2021), we use a conjugate normal distribution for the prior $P(\mu_{h(i)} \mid \mathcal{T}_h)$: $\mu_{h(i)} \mid \mathcal{T}_h \sim \mathcal{N}(0, \sigma_\mu^2)$. Then the model's estimate for a single subject is $\mu_i = \text{E}[y \mid \boldsymbol{x}_i] = \mu_0 + \sum_h^H \mu_{h(i)}$, where $\mu_i \sim \mathcal{N}(\mu_0, H\sigma_\mu^2)$. Here, we choose a value for σ_μ that satisfies $y_{\min} = \mu_0 - \kappa\sqrt{H}\sigma_\mu$ and $y_{\max} = \mu_0 + \kappa\sqrt{H}\sigma_\mu$ which is $\sigma_\mu = \frac{y_{\max} - y_{\min}}{2\kappa\sqrt{H}}$. So, for a single leaf. we arrive at the prior $\mu_{hl} \sim \mathcal{N}(0, [\frac{y_{\max} - y_{\min}}{2\kappa\sqrt{H}}]^2)$. As originally suggested by Chipman et al. (2010), setting $\kappa = 2$ ensures a 95% prior probability that $\text{E}[y \mid \hat{\boldsymbol{x}}_i]$ is $\in [y_{\min}, y_{\max}]$.

3.2.3 *Prior for standard deviation* $P(\sigma)$

As proposed in Chipman et al. (2010), we use the inverse chi-square distribution as a conjugate prior on the error variance, $\sigma^2 \sim \text{inverse-}\chi^2(\nu)$. Here, the hyperparameter ν is chosen by comparison to a functional linear regression fit to the data, with the idea being that a linear regression will overestimate the residual standard deviation. Hence, ν is chosen such that $P(\sigma < \hat{\sigma}_{\text{OLS}}) = 0.9$.

3.3 Posterior inference

For posterior inference, we rely on a Bayesian backfitting Markov chain Monte Carlo (MCMC) algorithm (Chipman et al., 2010). Here, we briefly review the sampling steps specific to the lsBART model, including efficient updates to the split probabilities and the application of a spatial smoothing kernel. We include details on the remaining sampling steps in Appendix A.

3.3.1 *Updating the predictor and interval probabilities*

We can efficiently update the probabilities of a predictor function, $\boldsymbol{\pi}$, and the probabilities of a location within a predictor function, $\boldsymbol{\tau}_j$, being chosen for a split. As noted in Linero (2018), the conjugate Dirichlet prior allows for Gibbs updates. The posterior full conditional distributions for $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_p\}$ and $\boldsymbol{\tau}_j = \{\tau_{j1}, \tau_{j2}, \ldots, \tau_{jS}\}$ are found by incorporating the number of times a tree chooses a particular predictor function and location within that function. Let $\boldsymbol{\Psi}$ represent all model parameters. Let c_j be the total count of the jth functional predictor variable being used as a splitting rule across all trees. Likewise, let c_{jk} be the total count of the kth location interval for the jth functional predictor being used as a splitting rule across all trees. We obtain the posterior full conditionals as the following:

$$\begin{aligned} \boldsymbol{\pi} \mid \boldsymbol{\Psi} &\sim \text{Dirichlet}(\alpha_{\pi_1} + c_1, \alpha_{\pi_2} + c_2, \ldots, \alpha_{\pi_p} + c_p), \\ \boldsymbol{\tau}_j \mid \boldsymbol{\Psi} &\sim \text{Dirichlet}(\alpha_{\tau_{j1}} + c_{j1}, \alpha_{\tau_{j2}} + c_{j2}, \ldots, \alpha_{\tau_{jS}} + c_{jS}). \end{aligned} \tag{1}$$

As noted in Linero (2018), these updates are in fact an approximation, based on the assumption that every predictor is a valid choice for every split. Since our predictors are continuous, we expect this to be generally the case.

3.3.2 *Smoothing of the* τ *probabilities*

For each MCMC draw, we update the neighbouring positions within $\boldsymbol{\tau}_j$ by using a Gaussian neighbourhood function. This smoothing process ensures that the model will consider neighbouring locations within a predictor function and hence, leverages the correlation across the domain of functional data. Here, we took inspiration from similar implementations of neighbourhood

functions for spatial correlation used in the self-organizing map algorithm (Kohonen, 1990). First, we calculate the Gaussian neighbourhood for the kth location probability, τ_{jk}, as $\tau_{jk} \times \exp\left(\frac{-(k-k')^2}{2\sigma_\tau^2}\right)$ for $k' = \{1, \ldots, S\}$, where σ_τ is the bandwidth parameter. We calculate this neighbourhood vector for each τ_{jk}. Then, we update the probabilities of each location by adding all of its Gaussian neighbourhood location probabilities. Formally, we update the kth location probability as

$$\tau_{jk} = \tau_{jk} + \sum_{k'=1}^{S} \tau_{jk'} \times \exp\left(\frac{-(k-k')^2}{2\sigma_\tau^2}\right), \tag{2}$$

Finally, we normalize the $\boldsymbol{\tau}_j$ vector, making sure $\sum_{k=1}^{S} \tau_{jk} = 1$. This smoothing step leverages the fact that functional data values are highly correlated to their neighbouring values on the underlying spatial grid. Hence, if a location s_k contains important information for the prediction task, then it is likely that both s_{k-1} and s_{k+1} will also contain important information. Thus, the trees will be encouraged to look at neighbouring values as well.

The novelty of our approach lies in smoothing the location probabilities described above and in the priors on the tree structures as described in Section 3.2. This allows lsBART to identify both relevant predictors and important regions within those predictors. Although the initial update of the location probabilities, $\boldsymbol{\tau}_j$, reflects a draw from a Dirichlet posterior full conditional, the subsequent smoothing process described in Equation (2) is an ad hoc modification rather than part of a formally defined Bayesian posterior. This smoothing step is introduced to capture the smooth spatial variation inherent in functional data, ensuring that neighbouring locations are considered jointly during prediction. Our approach allows us to balance the preference for sparsity expressed through the Dirichlet prior with a desire for smoothness, while maintaining computational efficiency. The approach of transforming MCMC draws from an unconstrained distribution to accommodate constraints is supported by the work of Yang et al. (2010) on semiparametric Bayes hierarchical models with mean and variance constraints. Yang et al. (2010) show that transforming the draws from an unconstrained Dirichlet process prior is more computationally convenient than sampling directly from a constrained posterior.

3.4 Binary classification

Bayesian additive regression trees have a natural extension to binary classification problems through the use of a probit model (Chipman et al., 2010). We can similarly extend lsBART to the binary case where $y = 0$ or $y = 1$. When predicting the probability that a single observation $\hat{\boldsymbol{x}}_i$ belongs to class $y = 1$, let

$$\mathrm{p}(\hat{\boldsymbol{x}}_i) \equiv P[y = 1 \mid \hat{\boldsymbol{x}}_i] = \Phi(\mu_0 + f(\hat{\boldsymbol{x}}_i)), \tag{3}$$

where Φ is the c.d.f. of the standard normal distribution. Unlike other ensemble tree classification algorithms, which use a majority or average vote from the trees, here, the classification probability is a function of the sum of trees output. Thus, the more negative that $\mu_0 + f(\hat{\boldsymbol{x}}_i)$ is, the closer the $p(\hat{\boldsymbol{x}}_i)$ is to 0. Likewise, the more positive, the closer $p(\hat{\boldsymbol{x}}_i)$ will be to 1. If all the tree values and the constant add up to 0, then $p(\hat{\boldsymbol{x}}_i) = 0.50$.

4 Results

4.1 Simulation study

To demonstrate the utility of our method, we compare its performance in a regression setting with multiple functional predictors to existing alternative methods. For the simulation study, we create a set of functional predictors that are linked with the response y via Friedman's five-dimensional test function, which is frequently used as a benchmark to test performance. First, we randomly generate 10 values from a standard normal distribution as the function realizations across 10 equally spaced points in the interval [0, 1]. We then fit these 10 values to a smooth function using B-splines with eight basis functions. This becomes the generating function $X_1(s)$ for $s \in [0, 1]$,

which will be used for all our simulated functions. With this smooth generating function $X_1(s)$, we create 2,000 simulated functional observations by extracting 100 data points at specific s intervals. To mimic how functional data is observed in a real-world setting, where the recorded measurement has some noise associated with it, we add Gaussian noise to each data point. This process results in 2,000 functional observations of 100 data points that are all noisy versions of $X_1(s)$. We repeat this process to get a total of five predictor functions: $X_1(s)$, $X_2(s)$, $X_3(s)$, $X_4(s)$, and $X_5(s)$.

We then compare our model's performance to alternative methods in a regression setting where y is generated using the Friedman function, which was also used in the original BART paper (Chipman et al., 2010; Friedman, 1991). The original function is

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5. \tag{4}$$

We modified the Friedman function to use functional data as inputs using specific s intervals on the underlying grid $s \in [0, 1]$, where for each predictor function X_j, we have an interval of discrete points s_j such that $s_j = \{s_{j1}, s_{j2}, \ldots, s_{jK}\}$:

$$y = \sum_{k=1}^{K}\Big(10\sin(\pi X_1(s_{1k})X_2(s_{2k})) + 20(X_3(s_{3k}) - 0.5)^2 + 10X_4(s_{4k}) + 5X_5(s_{5k})\Big), \tag{5}$$

To closely mirror real-world data conditions, the error variance for y was set to $\sigma = 74.9$, yielding a signal-to-noise ratio (SNR) of 11. This choice was informed by our motivating dataset, where an SNR of 11.4 was estimated using a logistic regression model (Czanner et al., 2008). For the simulation study, we chose the following intervals; $s_1 = \{0.01, 0.02, \ldots, 0.10\}$, $s_2 = \{0.21, 0.22, \ldots, 0.30\}$, $s_3 = \{0.41, 0.42, \ldots, 0.50\}$, $s_4 = \{0.61, 0.62, \ldots, 0.70\}$, and $s_5 = \{0.81, 0.82, \ldots, 0.90\}$. This is to illustrate that each predictor function has a unique window of importance in the regression. Location smoothed BART should detect these important locations through the $\boldsymbol{\tau}_j$ probability values of a given predictor function. To highlight the sparsity-inducing effects of the lsBART model, we include scenarios with additional noisy predictor functions that have no relation to the outcome y. We consider a range of settings from a relatively simple case with the five original predictor functions to a setting with 50 predictor functions, where 45 predictor functions are not used in the generation of y.

We evaluated lsBART against BART, DART, functional linear regression, LASSO regression, ridge regression, a feed-forward neural network with a single hidden layer, and the random forest model, using the `BART`, `fda`, `glmnet`, `nnet` and `randomForest` packages (Friedman et al., 2010; Liaw & Wiener, 2002; Ramsay et al., 2020; Sparapani et al., 2021; Venables & Ripley, 2002). The 2,000 observations were split evenly into training and test sets, with performance assessed using test set root mean squared error (RMSE). We extract 100 evenly spaced points on the underlying s grid so that $s = \{0.01, 0.02, \ldots, 1.00\}$ for each predictor function. Hence, for five predictor functions, we have $\hat{\mathbf{X}}_{1000\times(5*100)}$ as the predictor training matrix, while for the 50 predictor function case, we have $\hat{\mathbf{X}}_{1000\times(50*100)}$. We used default model parameters for all tree-based models, including 200 trees and 100 draws for lsBART, BART, DART, and random forest. We set the hyperparameter $\sigma_\tau = 1$ for lsBART. We used the same training and test matrices for each model.

We see in Figure 2 that the lsBART model, indicated by the solid line with triangular points, achieves a substantially lower RMSE than the competing methods, and that its performance is relatively unaffected by the inclusion of additional noise predictor functions. Averaged over all settings, lsBART had an RMSE of 105, while its closest competitor DART had an RMSE of 166, BART 188, random forest 204, LASSO 220, ridge regression 227, neural network 235, and functional regression 256. Additionally, the gap in performance increased with the number of predictor functions, indicating that lsBART is the most robust to the inclusion of irrelevant predictor functions. Furthermore, we found that the time it takes to train the model remains relatively stable, averaging at about 4.4 s, even with 50 functional predictors. This is the case for all BART models and most linear regression models, in stark contrast to the random forest, neural network, and ridge regression models, which require significantly more training time as the number of functional predictors increases. Chipman et al. (2010) found similar results when comparing the run time of BART to the random forest model.

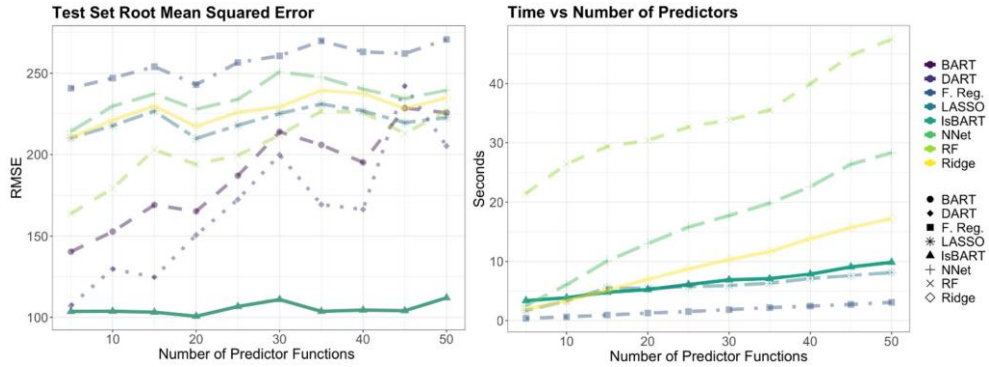

Figure 2. Comparing location smoothed Bayesian additive regression trees (lsBART) performance on the Friedman function in terms of root mean squared error (RMSE) on a test set (left) and time to train the model (right); BART and Dirichlet additive regression trees (DART) had similar run times to lsBART and are omitted in the right-hand plot for visual clarity.

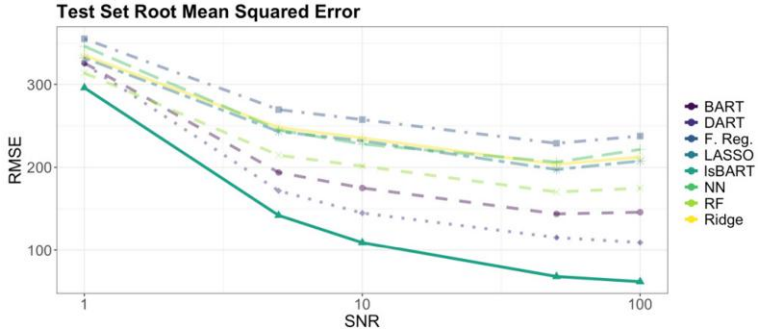

Figure 3. Comparing location smoothed Bayesian additive regression trees (lsBART) performance as the Signal-to-Noise ratio varies on 15 predictor functions.

4.1.1 Signal-to-noise ratio

We look at lsBART's performance as we alter the SNR in the simulation. Previously, we assessed the models' performance at an SNR of 11. For this simulation set-up, we used a total of 15 functional predictors where five functional predictors are used in the generation of y using the modified Friedman function from Equation (5), and 10 are predictor functions not used in the generation of y. Now, we examine how RMSE changes as we vary the SNR from 1 to 100, by altering the error variance on y. In Figure 3, we see that the lsBART model outperforms the other models across SNR values. Averaged over all SNR values, lsBART had an RMSE of 135, while DART had an RMSE of 173, BART 196, random forest 215, LASSO 242, ridge regression 247, neural network 249, and functional regression 269.

4.1.2 lsBART inference

Aside from outperforming the other models in prediction, our model is the only model that selects both relevant functions and regions within those functions. Figure 4 shows how the lsBART model is able to capture the dependence of y on the unique intervals in each predictor function. We illustrate the case with 10 predictor functions, where X_1 through X_5 are the predictor functions that contain signal, while X_6 through X_{10} are additional noisy predictor functions. Figure 4a shows the posterior $\boldsymbol{\pi}$ probabilities from the lsBART model. The posterior probabilities accurately indicate that X_1 through X_5 are the important predictor functions. This is particularly impressive as it captures both X_1 and X_2, which are put through the *sine* function and hence are difficult to

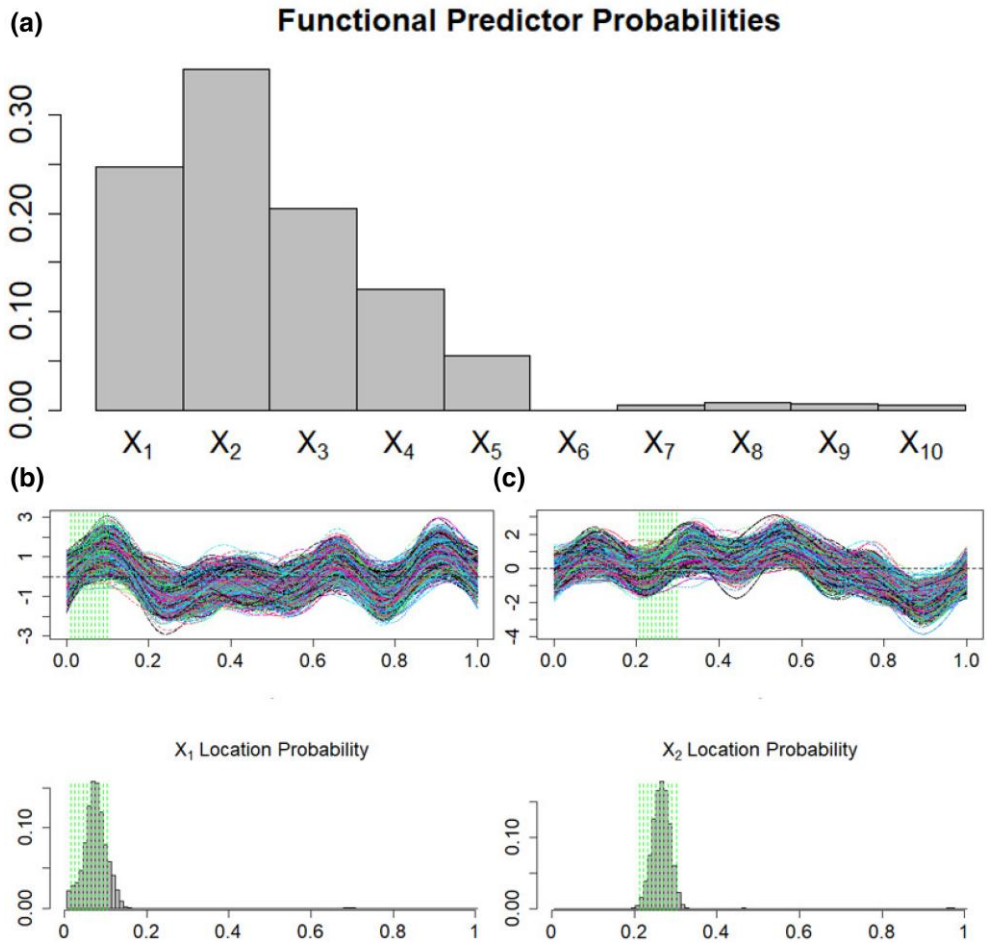

Figure 4. (a) The $\boldsymbol{\pi}$ probabilities from the location smoothed Bayesian additive regression trees (lsBART) model, which capture the five important functions and leave out the five noisy function predictors; (b) All $\mathbf{X}_1$ predictor functions and the $\boldsymbol{\tau}_1$ probabilities from $\mathbf{X}_1$, which captures the true important region highlighted in green; (c) Similarly shows $\mathbf{X}_2$ and the $\boldsymbol{\tau}_2$ probabilities for the third functional predictor $\mathbf{X}_2$, which correctly capture the true important region.

detect because of their nonlinear influence on y. Furthermore, we see that our proposed lsBART model is able to capture the important regions of each predictor function used in the modified Friedman function. In Figure 4b, we see a plot of X_1 with the region of importance, $s_1 = \{0.01, 0.02, \ldots, 0.10\}$, indicated with green lines. The $\boldsymbol{\tau}_1$ probabilities for X_1 accurately identify the truly relevant s_1 region. Similarly, in Figure 4c, we see a plot of X_2 with the region of importance, $s_2 = \{0.21, 0.22, \ldots, 0.30\}$, indicated with green lines. The $\boldsymbol{\tau}_2$ probabilities for X_2 capture that s_2 region. We found that all $\boldsymbol{\tau}_j$ probabilities accurately reflected the region of importance for the remaining predictor functions. Hence, using the lsBART model, a researcher is able to determine important functional predictors and unique regions within each function that are relevant to the outcome.

4.2 Application to radiation treatment planning

We obtained a set of kidney contours from a cohort of 140 patients who underwent radiation treatment planning as a part of therapy for cervical cancer. The kidney is contoured during the planning process as it constitutes a nearby organ at risk. Since most patients have two kidneys, this yields two structures per patient plan. The contours were generated by the Radiation Planning Assistant (RPA), using a deep learning model based on a convolutional neural network

(CNN) algorithm (Rhee, Jhingran, Rigaud, et al., 2020). In total, we obtained 260 contours that were deemed clinically acceptable. A dosimetrist then manually introduced errors, yielding 52 unacceptable contours. We applied the lsBART model to predict the binary response of whether a contour of the kidney is clinically acceptable or not.

From the axial view of the kidney structure, we derived quantitative shape statistics to capture its geometric aspects, including familiar summaries such as area and perimeter and also more advanced features such as sphericity, which describes how closely a shape resembles a sphere, and rectangularity, which describes how closely a shape resembles a rectangle (Dryden & Mardia, 2016; Rosin, 2005; Wirth, 2004). Leveraging these shape statistics obtained per slice allows us to apply the lsBART model to the stack of contours generated for each kidney. As each plan comprises stacked images, we obtained multiple shape statistics from each slice, resulting in a functional observation with s denoting the slice location. We relied on the binary matrix representation of the contour mask, which reflects a singular, closed contour. Edge voxels were used to define the perimeter. We relied on the `EBImage` package to compute sphericity based on the minimum, mean, maximum, and standard deviation of the radii lengths from the contour's midpoint (Pau et al., 2010). We utilized the `grDevices` package to identify the contour's convex hull, enabling convexity and roundness calculations (R Core Team, 2023). We computed area using the `concaveman` package (Gombin et al., 2020) and metrics, including circularity, eccentricity, elongation, rectangularity, and centroid size using the `Momocs` package (Bonhomme et al., 2014). These metrics, recorded per slice, produced a feature vector across slices, leading to a functional observation. In total, each kidney contour was summarized using 345 values, derived from 15 shape statistics across 23 location points.

We compared the performance of lsBART in terms of cross-validated classification accuracy to BART and DART, its closest competitors in the simulation study. We used 100 trees with 200 draws for each model, and set $\sigma_\tau = 2$ when applying lsBART. To obtain out-of-sample predictions for each method, we performed 10 replicates of 10-fold cross-validation. For each fold, we have $\hat{\mathbf{X}}_{280\times(15*23)}$ as our training set and $\hat{\mathbf{X}}_{32\times(15*23)}$ as our test set. Table B1 in Appendix B summarizes the relative performance of the models using comparison metrics, including the area under the curve (AUC) of the receiver operating characteristic (ROC) and precision-recall (PR) curves, sensitivity, and specificity. In Table B1, we see that lsBART performed similarly to BART and DART, achieving an AUC value of 0.905. Based on these cross-validation results, we see that the lsBART is a competitive option; as described in more detail in the next section, it provides an added bonus of interpretability that BART and DART do not.

4.2.1 Location interpretation with Shapley values

Shapley values in machine learning explain model predictions by evaluating the contribution of individual features for specific instances (Cohen et al., 2005; Shapley, 1953). Unlike global importance scores, Shapley values provide insights tailored to the model prediction for a specific input observation. Features that, when removed, significantly impact the prediction receive high Shapley values. To elaborate, let $A = \{x_1, x_2, \ldots, x_p\}$ denote the set of p input features. To determine how much each feature contributes to the function f, we evaluate the contribution of each feature x_j by calculating its Shapley value ϕ_j:

$$\phi_j = \sum_{B \subseteq A \setminus \{x_j\}} \frac{|B|!(p - |B| - 1)!}{p!} [f(B \cup \{x_j\}) - f(B)], \tag{6}$$

where B represents any subset of A that excludes x_j. The idea is to measure how the output of f changes when x_j is added to different subsets B, and then average this contribution over all possible subsets. Our model's location-specific nature allows us to use Shapley values to identify which functional predictors and which specific locations within these functions influence the prediction. The rapid tree-based Shapley approach, *TreeSHAP*, has been effectively integrated into models such as random forest and XGBoost. We incorporated this approach into the lsBART framework using the `treeshap` package (Komisarczyk et al., 2023). To the best of our knowledge, the application of Shapley values for interpretation in the BART framework appears novel. The unique design of BART, which introduces a new ensemble of trees with each MCMC iteration, enables us to

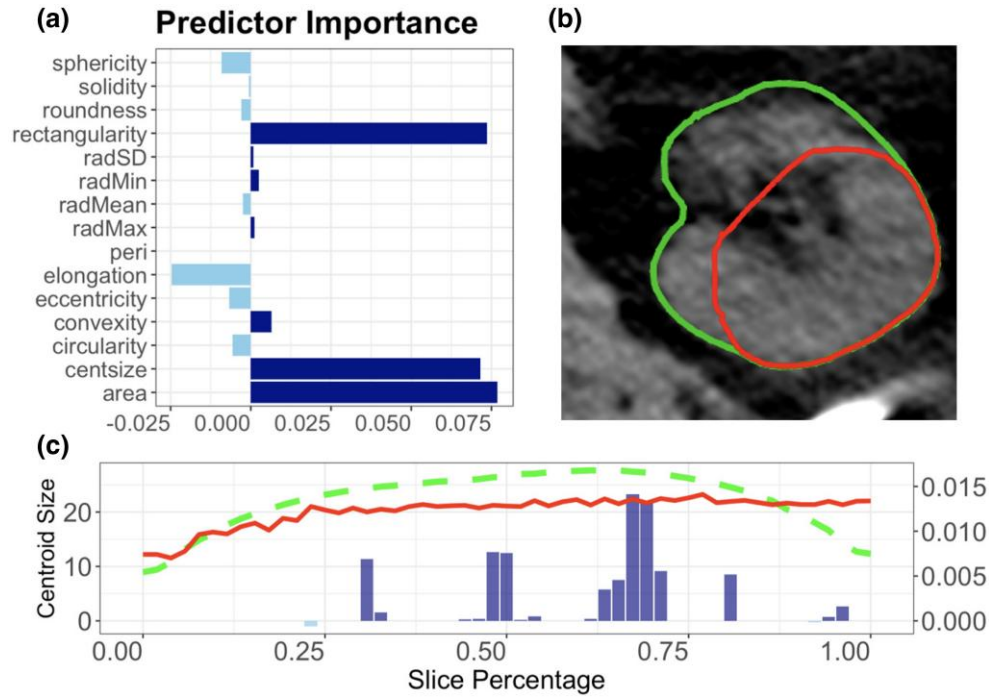

Figure 5. (a) The Shapley values for each functional predictor; (b) CT image showing the acceptable contour in green and the unacceptable contour in red; (c) the Shapley values of location within the centroid size predictor function in blue bars, the mean functional predictor of acceptable kidney contours in green, and the flagged kidney contour functional predictor in red.

compute Shapley values across each tree and iteration. Averaging these values yields a mean Shapley value for predictive features. For binary classifications, Shapley values are derived from the leaf values of $g(\hat{\boldsymbol{x}}_i, \mathcal{T}_b, \mathcal{M}_b)$ and not probabilities, due to BART's inherent sum of trees structure. To interpret the Shapley values in this case we recall that $\Phi(\mu_0 + f(\hat{\boldsymbol{x}}_i)) \to 0$ as $\mu_0 + f(\hat{\boldsymbol{x}}_i) \to -\infty$ and $\Phi(\mu_0 + f(\hat{\boldsymbol{x}}_i)) \to 1$ as $\mu_0 + f(\hat{\boldsymbol{x}}_i) \to \infty$. Hence, negative Shapley values contribute to the prediction being 0, and positive Shapley values contribute to the prediction being 1.

To illustrate the interpretability of the Shapley values obtained from lsBART, we performed an 80% training and 20% testing split of the kidney contour dataset and calculated the mean Shapley values from lsBART for an exemplary test set contour that was flagged (Figure 5). In Figure 5a, we see the Shapley values for each shape statistic, with rectangularity, centroid size, and area being of particular interest. Figure 5b shows an image from the 65% axial location within the treatment plan with both a clinically acceptable contour in green and an unacceptable contour in red. The unacceptable contour has a much smaller area and centroid size in comparison to the acceptable contour. Figure 5c shows the Shapley values for the location in the plan of the centroid size predictor function. We see a clear spike in the Shapley value from 60%–70%, indicating something suspicious in that region. Furthermore, we show the average centroid size predictor function of all the acceptable kidney contours in green and the current flagged observation's centroid size function in red. The large spike in the Shapley value corresponds to a large difference in the flagged contour's centroid size value from the mean of the acceptable contours. Hence, we see that the largest Shapley value spike indicates the location of a specific error in the plan. We show the patient-specific interpretation of a potential contouring error with another example in Figure 6. As in Figure 5, here we see an axial slice from the treatment plan with both the acceptable (green) and unacceptable (red) contours. The image in Figure 6b is from about the 55% location of the images in the plan and corresponds to the location of the largest Shapley value from the rectangularity predictor function. We are thus able to simultaneously flag a plan and find which images in the plan contain the error.

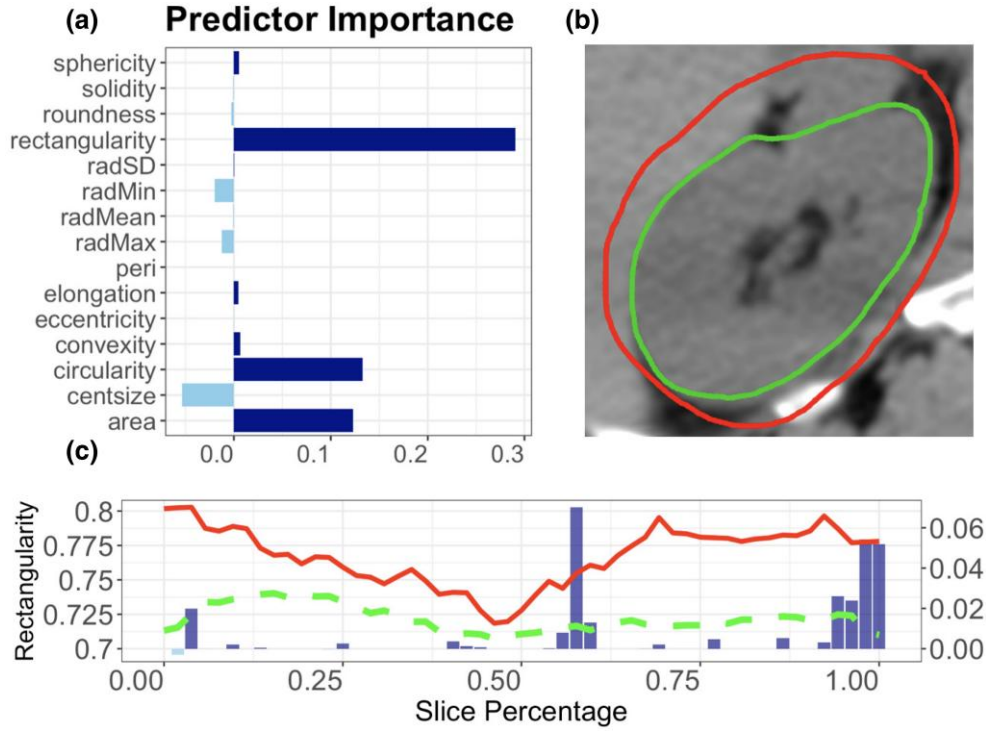

Figure 6. (a) The Shapley values for each functional predictor; (b) CT image showing the acceptable contour in green and the unacceptable contour in red; (c) the Shapley values of location within the rectangularity function in blue bars, the mean functional predictor of acceptable kidney contours in green, and the flagged kidney contour functional predictor in red.

4.2.2 Application to unlabelled data

We aimed to evaluate the effectiveness of our innovative method on an external dataset. Initially, we trained the lsBART model on the comprehensive dataset of 312 kidney contours. Since a more sensitive classifier is desired in the context of radiation therapy quality assurance to ensure meticulous review of suspicious cases, we identified an optimized cut-off threshold using Youden's index. Training on the full dataset, lsBART with Youden's index achieved a total accuracy of 93.27%, AUC_{ROC} value of 0.983, and AUC_{PR} value of 0.945.

We acquired an external data set of 18 radiation treatment plans for cervical cancer radiotherapy from a radiation physics lab at MD Anderson Cancer Center. From these, we extracted 36 kidney contours that were computer generated. As these contours were novel to our model, they were treated as unlabelled data. Following the extraction of shape feature functional predictors as earlier described, we applied our trained lsBART model to obtain interpretable predictions. Figure 7 shows the estimated probabilities with standard deviations from each posterior draw of each contour being unacceptable for use in radiotherapy planning. Notably, nine contours were flagged as surpassing Youden's index threshold, the black dashed horizontal line. As would happen in the potential clinical application of our approach, a dosimetrist then inspected the flagged contours to simulate the clinical workflow. Of these, seven were discerned to have discrepancies, ranging from over-contouring to under-contouring. Notably, the remaining two flagged contours, while error-free, were truncated, affecting a segment of the kidney and possibly leading to their high posterior probabilities.

The dosimetrist subsequently reviewed the remaining contours to identify ones that would be clinically unacceptable. In total, out of the 36 contours, 15 exhibited errors that rendered them clinically unacceptable, whereas 21 would be acceptable. Using Youden's index from the training set, we achieved a total accuracy of 72.2% and AUC_{ROC} and AUC_{PR} values of 0.641 and 0.628, respectively.

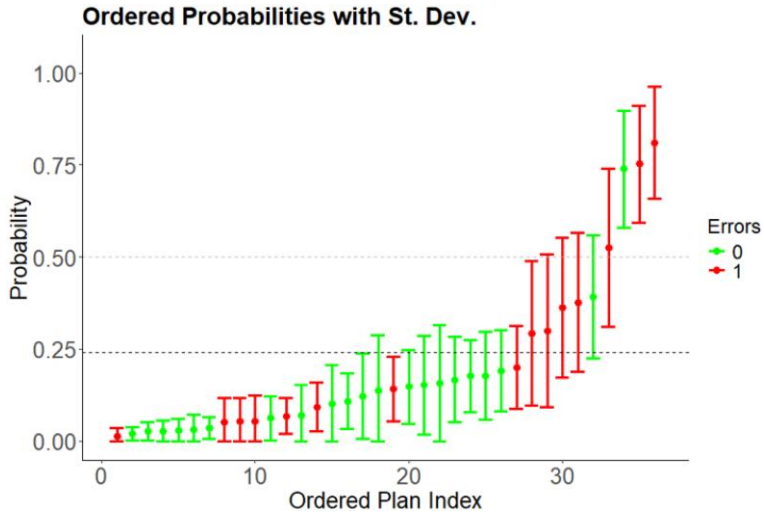

Figure 7. Probabilities from the external dataset where the unacceptable contours are in red and the acceptable contours are in green, with probabilities ordered from lowest to highest.

A pivotal aspect of our advanced quality assurance methodology is the interpretability offered by integrating Shapley values with lsBART. When contours are flagged, we can derive useful explanations for the flagging and pinpoint suspicious regions within the plan. In Figure 8, the Shapley values for each functional predictor are illustrated, accompanied by an axial-view image of a kidney contour from the position with the largest Shapley value. These Shapley values, associated with functional predictors, provide insights into potential locations of error. As observed, a pronounced peak of the Shapely value on a functional predictor consistently correlates with the detected anomaly. Additional illustrations of flagged errors are provided in Appendix B.

5 Discussion

Location smoothed Bayesian additive regression trees are a novel extension of BART that prioritizes identifying important predictor functions and their significant regions. The motivating application is for locating contouring errors within treatment plans. To the best of our knowledge, we are first in applying a functional data framework for radiation treatment plan quality assurance and in integrating Shapley values within a BART framework.

In our model, we encourage smoothness of the location probabilities across space. To ensure sparsity and maintain computational efficiency, we achieve this by imposing a constraint on the Dirichlet prior for the location parameters, which we enforce by manipulating our posterior samples. Thus, our algorithm can be considered approximate. We also considered alternative fully Bayesian formulations for achieving smoothness, such as placing a Gaussian process prior on $\boldsymbol{\tau}_j$. We found that this approach resulted in denser models with increased computational time. Further exploration of a fully Bayesian approach to lsBART remains a direction for future work.

However, our study is not without limitations. Firstly, the unacceptable contours in the training data were manually generated, since any contours where errors are noticed would be corrected before use in clinical radiotherapy treatment planning, and are therefore difficult to obtain. This manual generation process may not fully represent all potential clinical scenarios. Secondly, given the heterogeneous nature of contour errors, a supervised classifier could miss a contour error that reflects an issue not seen in the training data. Future contour quality assurance methods could incorporate more deep learning techniques. With the emergence of vision transformer models, such as GPT-4V, the focus could shift significantly towards deep learning image processing. However, privacy concerns and inconsistencies in medical image interpretation must be addressed to ensure these models' reliability and accuracy in clinical practice (OpenAI, 2023).

Our work can be extended to address other settings beyond radiation treatment planning with a need for classifying objects. While lsBART currently focuses on extracted shape statistics rather

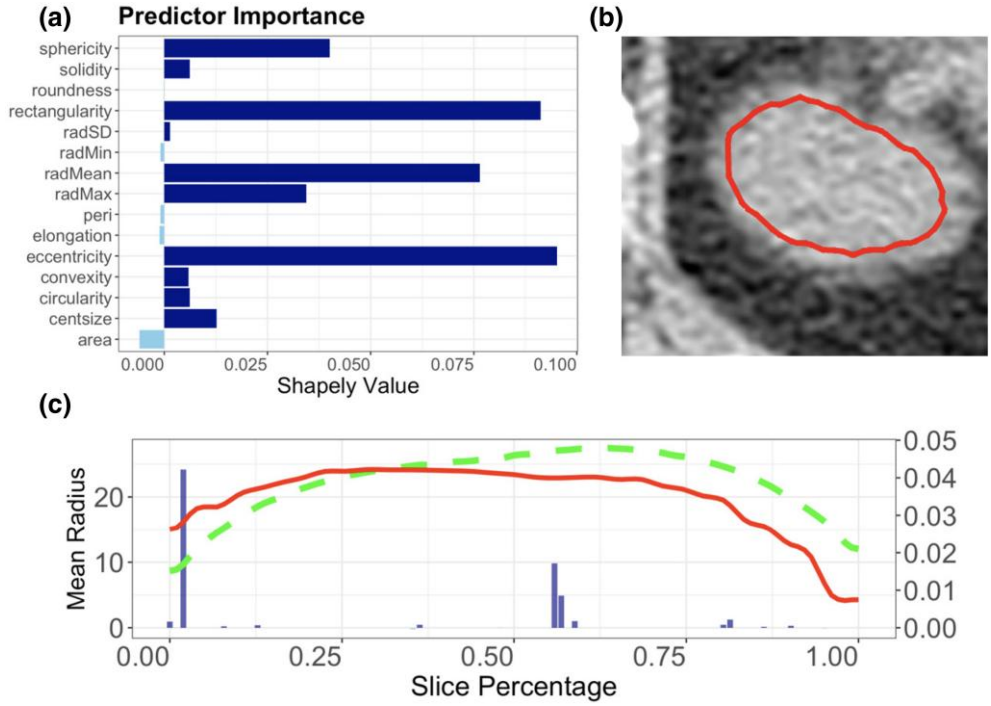

Figure 8. (a) The Shapley values for each functional predictor; (b) CT image showing the unlabelled contour in red; (c) the Shapley values of location within the Mean Radius function in blue bars, the mean functional predictor of acceptable kidney contours in green, and the flagged kidney contour functional predictor in grey.

than direct image analysis, potential applications include extending the methodology to predict treatment outcomes directly from segmented MRI images of tumours or applying it to computer vision tasks such as object detection and classification in autonomous driving. More broadly, our lsBART model can be generalized to other prediction tasks with multiple functional or spatially related inputs, including auditory signal processing and applications in public health such as associating disease risk with time-varying exposures to environmental pollutants. A potential application in genomics would be outcome prediction from methylation data; this can be seen as a prediction problem with a functional predictor for each chromosome. Methylation markers relevant to the outcome are likely to be concentrated within spatial regions, corresponding to the nearby genes that they regulate. We expect that our proposed model would work well in this setting.

Conflicts of interest: L.E.C. is a Scientific Advisory Board member for Leo Cancer Care.

Funding

Z.T.W. was partially supported by the National Institute of Health and the National Cancer Institute training grant T32 CA096520 and the National Science Foundation GRFP Grant No. 1842494. C.B.P. was partially supported by the National Institute of Heath and the National Cancer Institute CCSG P30 CA016672 (Biostatistics Resource Group) and by a grant from Varian Medical Systems. L.E.C. effort on this project was funded by a grant from Varian Medical Systems, along with other funding from the Cancer Prevention and Research Institute of Texas and the Fund for Innovation in Cancer Informatics.

Data availability

We provide code to run the lsBART algorithm and to reproduce the analysis. We also provide the shape statistic functional predictors from the labelled and unlabelled kidney data sets. The code

and datasets are available online here: https://github.com/wootz101/LocationSmoothedBART. Moreover, we would like to thank Licai Huang for guidance on the BART code.

Appendix A

We now explain in full detail the posterior distribution and how to update the model through each posterior sample. After describing the updates for each parameter, we summarize the sampling steps in Algorithm 1.

A.1 Posterior calculations

Let $\boldsymbol{\Psi}$ represent all model parameters, and let D represent all of the data, so $D = \{(X_1(s), y_1), (X_2(s), y_2), \ldots, (X_n(s), y_n)\}$. Then, we can define the posterior as proportional to the likelihood times the prior distributions.

$$P(\boldsymbol{\Psi} \mid D) \propto \prod_{i=1}^{n} \left\{ \prod_{h=1}^{H} P(y_i \mid \hat{X}, \mathcal{T}_h, \mathcal{M}_h, \sigma) P(\mathcal{M}_h \mid \mathcal{T}_h) P(\mathcal{T}_h) P(\sigma) \right\}. \tag{A1}$$

A.2 Updating $\mathcal{T}_t$, $\mathcal{M}_t$, and σ^2

We can use the residuals to update a single tree at a time (Chipman et al., 2010). We define the residual for the ith observation when tree t is left out as $R_{ti} = y_i - \sum_{h \neq t}^{H} g(\hat{\boldsymbol{x}}_i, \mathcal{T}_h, \mathcal{M}_h) = \mu_0 + g(\hat{\boldsymbol{x}}_i, \mathcal{T}_t, \mathcal{M}_t) + \varepsilon_i$ and hence we see that $R_{ti} \sim N(\mu_0 + g(\hat{\boldsymbol{x}}_i, \mathcal{T}_t, \mathcal{M}_t), \sigma^2)$.

A.3 Updating terminal nodes (leaves)

First, we note that $P(R_{ti} \mid \hat{\boldsymbol{x}}_i, \mathcal{T}_t, \mathcal{M}_t, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(R_{ti} - g(\hat{\boldsymbol{x}}_i, \mathcal{T}_t, \mathcal{M}_t))^2}{2\sigma^2}\right)$ and we recall that for the tth tree and lth leaf node $\mu_{tl} \mid \mathcal{T}_t \sim N(0, \sigma_\mu^2)$ and hence $P(\mu_{tl} \mid \mathcal{T}_t) = \frac{1}{\sigma_\mu\sqrt{2\pi}} \exp\left(-\frac{(\mu_{tl})^2}{2\sigma_\mu^2}\right)$.

Algorithm 1 Location smoothed BART Algorithm

Require: data format as $\hat{\boldsymbol{X}}_{n\times(p*S)}$ with scalar response, $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$, V: the number of iterations

for $v \leq V$ **do**
  for $h \leq H$ **do**
    $\mathcal{T}_h | R_h, \sigma^2$ ▷ update tree structure using Backfitting (Algorithm 2)
    $\mathcal{M}_h | R_h, \mathcal{T}_h, \sigma^2$ ▷ update leaf node values using Backfitting (Algorithm 2)
  end for
  $\boldsymbol{\pi} | \mathcal{T}_1, \mathcal{M}_1, \mathcal{T}_2, \mathcal{M}_2, \ldots, \mathcal{T}_H, \mathcal{M}_H$ ▷ draw predictor probabilities from posterior
  for $j \leq P$ **do**
    $\boldsymbol{\tau}_j | \boldsymbol{\pi}, \mathcal{T}_1, \mathcal{M}_1, \ldots, \mathcal{T}_H, \mathcal{M}_H$ ▷ draw location probabilities from posterior
  end for
  for $k \leq S$ **do**
    $\tau_{jk} = \tau_{jk} + \sum_{k'=1}^{S} \tau_{jk} \times \exp\left(\frac{-(k-k')^2}{2\sigma_\tau^2}\right)$ ▷ Gaussian smoothing
  end for
  for $j \leq P$ **do**
    $\boldsymbol{\tau}_j = \frac{\boldsymbol{\tau}_j}{\sum_{k=1}^{S} \tau_{jk}}$ ▷ Normalize location probabilities
  end for
  $\sigma^2 | \mathcal{T}_1, \mathcal{M}_1, \mathcal{T}_2, \mathcal{M}_2, \ldots, \mathcal{T}_H, \mathcal{M}_H$ ▷ update variance from posterior
end for

Algorithm 2 Backfitting Algorithm

Require data format as $\hat{\boldsymbol{X}}_{n\times(p*S)}$

for $1 \leq h \leq H$

  Propose $\mathcal{T}_h^* \sim q(\mathcal{T}_h^*; \mathcal{T}_h)$ ▷ Proposal can grow, prune, swap, change tree

  Let $a = \frac{L(\mathcal{T}_h^*; \mathcal{T}_{(h)}, \mathcal{M}_{(h)}, \boldsymbol{\theta}) P(\mathcal{T}_h^*)}{L(\mathcal{T}_h; \mathcal{T}_{(h)}, \mathcal{M}_{(h)}, \boldsymbol{\theta}) P(\mathcal{T}_h^*)} \frac{q(\mathcal{T}_h^*; \mathcal{T}_h)}{q(\mathcal{T}_h; \mathcal{T}_h^*)}$

  Set $\mathcal{T}_h = \mathcal{T}_h^*$ with probability $\min(a, 1)$

  Sample $M_h \sim P(\mathcal{M}_h | \mathcal{T}_h, \mathcal{T}_{(h)}, \mathcal{M}_{(h)}, \boldsymbol{\theta}, D)$ ▷ Draw new leaves

end for

Let $Y_{\mu_{tl}} = \{y_i : g(\hat{\boldsymbol{x}}_i, \mathcal{T}_t, \mathcal{M}_t) = \mu_{tl}\}$ be a $n_{\mu_{tl}}$ sized vector and the set of all observations that end in the terminal node μ_{tl}. Then the posterior distribution of μ_{tl} is

$$P(\mu_{tl} \mid \boldsymbol{\Psi}) \propto \left\{ \prod_{i=1}^{n_{\mu_{tl}}} P(R_{ti} \mid \hat{\boldsymbol{x}}_i, \mathcal{T}_t, \mathcal{M}_t, \sigma) \right\} P(\mu_{tl} \mid \mathcal{T}_t), \tag{A2}$$

$$P(\mu_{tl} \mid \boldsymbol{\Psi}) \sim \mathcal{N}\left(\frac{\sigma_\mu^2 \sum_{i=1}^{n_{\mu_{tl}}} R_{ti}}{\sigma_\mu^2 n_{\mu_{tl}} + \sigma^2}, \frac{\sigma^2 \sigma_\mu^2}{(\sigma_\mu^2 n_{\mu_{tl}} + \sigma^2)} \right). \tag{A3}$$

A.4 Updating the variance σ^2

We use a conjugate prior for the variance, where $\sigma^2 \sim$ inverse-$\chi^2(v)$, as mentioned in Section 3.

We obtain the posterior distribution for σ^2 as the following:

$$P(\sigma^2 \mid \boldsymbol{\Psi}) = \prod_{i=1}^{n} \{P(R_{ti} \mid \hat{\boldsymbol{x}}_i, \mathcal{T}, \mathcal{M}, \sigma^2)\} P(\sigma^2), \tag{A4}$$

$$P(\sigma^2 \mid \boldsymbol{\Psi}) \sim \text{inverse-Gamma}\left(\frac{v+n}{2}, \frac{4 + \sum_{i=1}^{n} (R_{ti} - g(\hat{\boldsymbol{x}}_i, \mathcal{T}_t, \mathcal{M}_t))^2}{2} \right). \tag{A5}$$

A.5 Updating the tree structure

Let $\mathcal{T}_{(h)} = \{\mathcal{T}_l : 1 \leq l \leq H, l \neq h\}$ be the set of all trees not including the hth tree. Likewise, let $M_{(h)} = \{M_l : 1 \leq l \leq H, l \neq h\}$ be the set of all mean leaf values not including the hth one. Let $\boldsymbol{\theta}$ be the vector of other parameters including σ, $\boldsymbol{\pi}$, and $\boldsymbol{\tau}$. Then the likelihood of the $\mathcal{T}_h$ tree is:

$$L(\mathcal{T}_h; \mathcal{T}_{(h)}, \mathcal{M}_{(h)}, \theta) = \int \left(\prod_{i=1}^{n} P(y_i \mid \mathcal{T}_h, \mathcal{M}_h, \mathcal{T}_{(h)}, \mathcal{M}_{(h)}, \boldsymbol{\theta}) \right) P(\mathcal{M}_h \mid \mathcal{T}_h, \boldsymbol{\theta}) d\mathcal{M}_h, \tag{A6}$$

Appendix B

In this section, we provide a summary of crosst-validated performance on the training data and additional examples of contour errors from the unlabelled data set (Figures B1 and B2).

Table B1. Quality assurance performance comparison based on cross-validated classification accuracy

	lsBART	BART	DART
AUC_{ROC}	0.905 (0.016)	0.908 (0.014)	0.898 (0.010)
AUC_{PR}	0.736 (0.033)	0.746 (0.018)	0.749 (0.023)
Sensitivity	0.294 (0.046)	0.288 (0.043)	0.390 (0.047)
Specificity	0.997 (0.004)	0.998 (0.002)	0.996 (0.003)

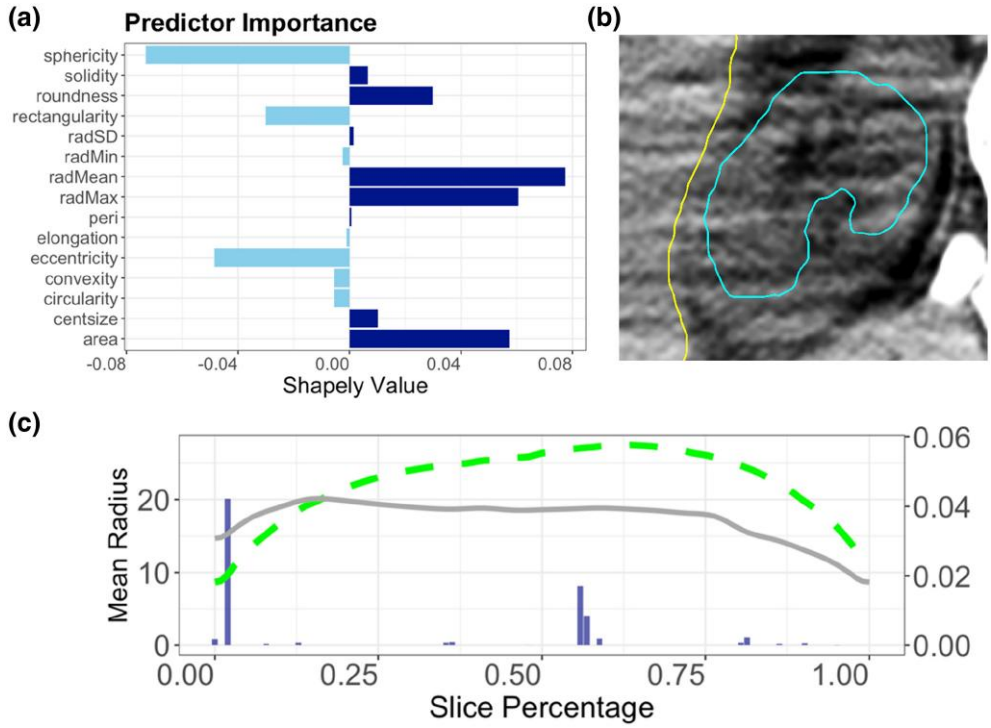

Figure B1. (a) The Shapley values for each functional predictor; (b) CT image showing the unlabelled contour in blue; (c) the Shapley values of location within the Mean Radius function in blue bars, the mean functional predictor of acceptable kidney contours in green, and the flagged kidney contour functional predictor in grey.

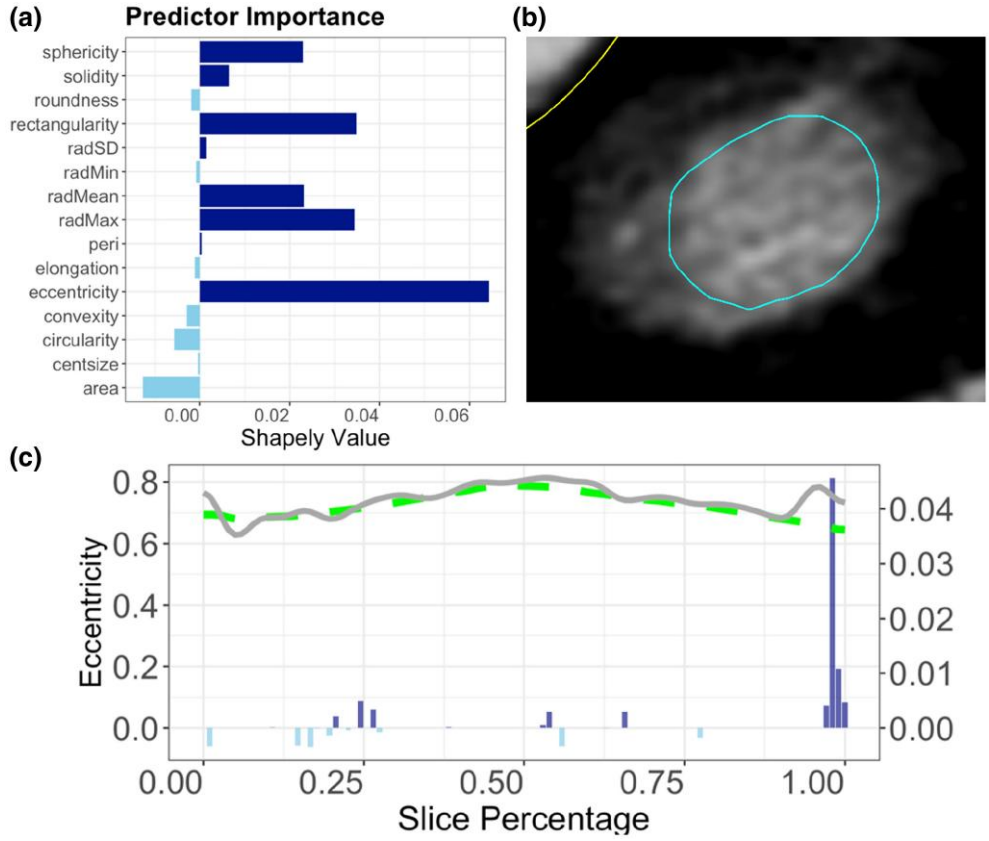

Figure B2. (a) The Shapley values for each functional predictor; (b) CT image showing the unlabelled contour in blue; (c) the Shapley values of location within the Eccentricity function in blue bars, the mean functional predictor of acceptable kidney contours in green, and the flagged kidney contour functional predictor in grey.

References

Bonhomme V., Picq S., Gaucherel C., & Claude J. (2014). Momocs: Outline analysis using R. *Journal of Statistical Software*, *56*(13), 1–24. https://doi.org/10.18637/jss.v056.i13

Breiman L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cardenas C. E., McCarroll R. E., Court L. E., Elgohari B. A., Elhalawani H., Fuller C. D., Kamal M. J., Meheissen M. A. M., Mohamed A. S. R., Rao A., Williams B., Wong A., Yang J., & Aristophanous M. (2018). Deep learning algorithm for auto-delineation of high-risk oropharyngeal clinical target volumes with built-in dice similarity coefficient parameter optimization function. *International Journal of Radiation Oncology, Biology, Physics*, *101*(2), 468–478. https://doi.org/10.1016/j.ijrobp.2018.01.114

Chen H.-C., Tan J., Dolly S., Kavanaugh J., Anastasio M. A., Low D. A., Harold Li H., Altman M., Gay H., Thorstad W. L., Mutic S., & Li H. (2015). Automated contouring error detection based on supervised geometric attribute distribution models for radiation therapy: A general strategy. *Medical Physics*, *42*(2), 1048–1059.https://doi.org/10.1118/1.4906197

Chipman H. A., George E. I., & McCulloch R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1), 266–298. https://doi.org/10.1214/09-AOAS285

Cohen S. B., Ruppin E., & Dror G. (2005). Feature selection based on the Shapley value. In *International Joint Conference on Artificial Intelligence*. https://api.semanticscholar.org/CorpusID:13081232.

Court L., Aggarwal A., Burger H., Cardenas C., Chung C., Douglas R., du Toit M., Jaffray D., Jhingran A., Mejia M., Mumme R., Muya S., Naidoo K., Ndumbalo J., Nealon K., Netherton T., Nguyen C., Olanrewaju N., Parkes J., … Beadle B. M. (2023). Addressing the global expertise gap in radiation oncology: The radiation planning assistant. *JCO Global Oncology*, *9*(9), e2200431. https://doi.org/10.1200/GO.22.00431

Czanner G., Sarma S., Eden U., & Brown E. (2008). A signal-to-noise ratio estimator for generalized linear model systems. *Lecture Notes in Engineering and Computer Science*, *2171*(1), 1063–1069.

Dryden I. L., & Mardia K. V. (2016). *Statistical shape analysis with applications in R*. John Wiley & Sons.
Du J., & Linero A. R. (2019). Incorporating grouping information into Bayesian decision tree ensembles. In *Proceedings of the 36th International Conference on Machine Learning*. https://par.nsf.gov/biblio/10097145.
Friedman J., Hastie T., & Tibshirani R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22. https://doi.org/10.18637/jss.v033.i01
Friedman J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, *19*(1), 1–67. https://doi.org/10.1214/aos/1176347963
Gombin J., Vaidyanathan R., & Agafonkin V. (2020). concaveman: A very fast 2D concave hull algorithm. R package version 1.1.0. https://CRAN.R-project.org/package=concaveman.
Kohonen T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480. https://doi.org/10.1109/5.58325
Komisarczyk K., Kozminski P., Maksymiuk S., & Biecek P. (2023). treeshap: Fast SHAP values computation for tree ensemble models. R package version 0.1.1.9001. https://github.com/ModelOriented/treeshap.
Liaw A., & Wiener M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. https://journal.r-project.org/articles/RN-2002-022/
Linero A. R. (2018). Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, *113*(522), 626–636. https://doi.org/10.1080/01621459.2016.1264957
McIntosh C., Svistoun I., & Purdie T. G. (2013). Groupwise conditional random forests for automatic shape classification and contour quality assessment in radiotherapy planning. *IEEE Transactions on Medical Imaging*, *32*(6), 1043–1057. https://doi.org/10.1109/TMI.2013.2251421
OpenAI (2023). GPT-4V(ision) System Card. https://cdn.openai.com/papers/GPTV_System_Card.pdf
Pau G., Fuchs F., Sklyar O., Boutros M., & Huber W. (2010). EBImage—an R package for image processing with applications to cellular phenotypes. *Bioinformatics*, *26*(7), 979–981. https://doi.org/10.1093/bioinformatics/btq046
Ramsay J. O., Graves S., & Hooker G. (2020). fda: Functional data analysis. R package version 5.1.4. https://CRAN.R-project.org/package=fda.
Ramsay J. O., & Silverman B. W. (2005). Principal components analysis for functional data. In *Functional Data Analysis* (pp. 147–172). Springer.
R Core Team (2023). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. https://www.R-project.org/.
Rhee D. J., Cardenas C. E., Elhalawani H., McCarroll R., Zhang L., Yang J., Garden A. S., Peterson C. B., Beadle B. M., & Court L. E. (2019). Automatic detection of contouring errors using convolutional neural networks. *Medical Physics*, *46*(11), 5086–5097. https://doi.org/10.1002/mp.v46.11
Rhee D. J., Jhingran A., Rigaud B., Netherton T., Cardenas C. E., Zhang L., Vedam S., Kry S., Brock K. K., Shaw W., O'Reilly F., Parkes J., Burger H., Fakie N., Trauernicht C., Simonds H., & Court L. E. (2020). Automatic contouring system for cervical cancer using convolutional neural networks. *Medical Physics*, *47*(11), 5648–5658. https://doi.org/10.1002/mp.14467
Rosin P. L. (2005). Computing global shape measures. In *Handbook of pattern recognition and computer vision* (pp. 177–196). World Scientific. https://doi.org/10.1142/9789812775320_0010
Shapley L. S. (1953). A value for *n*-person games. In *Contributions to the theory of games II* (pp. 307–317). Princeton University Press.
Sparapani R., Spanbauer C., & McCulloch R. E. (2021). Nonparametric machine learning and efficient computation with Bayesian additive regression trees: The BART R package. *Journal of Statistical Software*, *97*(1), 1–66. https://doi.org/10.18637/jss.v097.i01
Starling J., Murray J., Carvalho C., Bukowski R., & Scott J. (2020). BART with targeted smoothing: An analysis of patient-specific stillbirth risk. *The Annals of Applied Statistics*, *14*(1), 28–50. https://doi.org/10.1214/19-AOAS1268
Venables W. N., & Ripley B. D. (2002). *Modern applied statistics with S* (4th ed.). Springer.
Wirth M. A. (2004). Shape Analysis & Measurement. University of Guelph, Computing and Information Science Image Processing Group.
Wooten Z. T., Yu C., Court L. E., & Peterson C. B. (2023). Predictive modeling using shape statistics for interpretable and robust quality assurance of automated contours in radiation treatment planning. In *Pacific Symposium on Biocomputing 2023*. World Scientific. https://doi.org/10.1142/9789811270611_0036
Yang M., Dunson D. B., & Baird D. (2010). Semiparametric Bayes hierarchical models with mean and variance constraints. *Computational Statistics & Data Analysis*, *54*(9), 2172–2186. https://doi.org/10.1016/j.csda.2010.03.025