Journal of Machine Learning Research 23 (2022) 1-43 Submitted 6/21; Revised 10/22; Published 11/22

Tree-Values: Selective Inference for Regression Trees

Anna C. Neufeld ANEUFELD@UW.EDU
Department of Statistics
University of Washington
Seattle, WA 98195, USA

Lucy L. Gao LUCY.GAO@STAT.UBC.CA
Department of Statistics
University of British Columbia
Vancouver, British Columbia, V6T 1Z4, Canada

Daniela M. Witten DWITTEN@UW.EDU
Departments of Statistics and Biostatistics
University of Washington
Seattle, WA 98195, USA

Editor: Garvesh Raskutti

Abstract

We consider conducting inference on the output of the Classification and Regression Tree (CART) (Breiman et al., 1984) algorithm. A naive approach to inference that does not account for the fact that the tree was estimated from the data will not achieve standard guarantees, such as Type 1 error rate control and nominal coverage. Thus, we propose a selective inference framework for conducting inference on a fitted CART tree. In a nutshell, we condition on the fact that the tree was estimated from the data. We propose a test for the difference in the mean response between a pair of terminal nodes that controls the selective Type 1 error rate, and a confidence interval for the mean response within a single terminal node that attains the nominal selective coverage. Efficient algorithms for computing the necessary conditioning sets are provided. We apply these methods in simulation and to a dataset involving the association between portion control interventions and caloric intake.

Keywords: Regression trees, CART, selective inference, post-selection inference, hypothesis testing.

1. Introduction

Regression tree algorithms recursively partition covariate space using binary splits to obtain regions that are maximally homogeneous with respect to a continuous response. The Classification and Regression Tree (CART; Breiman et al. 1984) proposal, which involves growing a large tree and then pruning it back, is by far the most popular of these algorithms.

The regions defined by the splits in a fitted CART tree induce a piecewise constant regression model where the predicted response within each region is the mean of the observations in that region. CART is popular in large part because it is highly interpretable; someone without technical expertise can easily "read" the tree to make predictions, and to understand why a certain prediction is made. However, its interpretability belies the fact

that CART trees are highly unstable: a small change to the training dataset can drastically change the structure of the fitted tree. In the absence of an established notion of statistical significance associated with a given split in the tree, it is hard for a practitioner to know whether they are interpreting signal or noise. In this paper, we use the framework of selective inference to fill this gap by providing a toolkit to conduct inference on hypotheses motivated by the output of the CART algorithm.

Given a CART tree, consider testing for a difference in the mean response of the regions resulting from a binary split. A very naive approach, such as a two-sample Z-test, that does not account for the fact that the regions were themselves estimated from the data will fail to control the selective Type 1 error rate: the probability of rejecting a true null hypothesis, given that we decided to test it (Fithian et al., 2014). Similarly, a naive Z-interval for the mean response in a region will not attain nominal selective coverage: the probability that the interval covers the parameter, given that we chose to construct it.

In fact, approaches for conducting inference on the output of a regression tree are quite limited. Sample splitting involves fitting a CART tree using a subset of the observations, which will naturally lead to an inferior tree to the one resulting from all of the observations, and thus is unsatisfactory in many applied settings; see Athey and Imbens (2016). Wager and Walther (2015) develop convergence guarantees for unpruned CART trees that can be leveraged to build confidence intervals for the mean response within a region; however, they do not provide finite-sample results and cannot accommodate pruning. Loh et al. (2016) and Loh et al. (2019) develop bootstrap calibration procedures that attempt to provide confidence intervals for the regions of a regression tree. In Appendix A, we show that this bootstrap calibration approach fails to provide intervals that achieve nominal coverage for the parameters of interest in this paper.

As an alternative to performing inference on a CART tree, one could turn to the conditional inference tree (CTree) framework of Hothorn et al. (2006). This framework uses a different tree-growing algorithm than CART, and at each split tests for linear association between the split covariate and the response. As summarized in Loh (2014), the CTree framework alleviates issues with instability and variable selection bias associated with CART. Despite these advantages, CTree remains far less widely-used than CART. Furthermore, while CTree attaches a notion of statistical significance to each split in a tree, it does not directly allow for inference on the mean response within a region or the difference in mean response between two regions. Finally, while the CTree framework requires few assumptions, its inference is based on asymptotics.

In this paper, we introduce a finite-sample selective inference (Fithian et al., 2014) framework for the difference between the mean responses in two regions, and for the mean response in a single region, in a pruned or unpruned CART tree. We condition on the event that CART yields a particular set of regions, and thereby achieve selective Type 1 error rate control as well as nominal selective coverage.

The rest of this paper is organized as follows. In Section 2, we review the CART algorithm, and briefly define some key ideas in selective inference. In Section 3, we present our proposal for selective inference on the regions estimated via CART. We show that the necessary conditioning sets can be efficiently computed in Section 4. In Section 5 we compare our framework to sample splitting and CTree via simulation. In Section 6 we

compare our framework to CTree on data from the Box Lunch Study. The discussion is in Section 7. Technical details are relegated to the supplementary materials.

2. Background

2.1 Notation for Regression Trees

Given p covariates $(X_1, \ldots, X_p)$ measured on each of n observations $(x_1, \ldots, x_n)$, let $x_{j,(s)}$ denote the sth order statistic of the jth covariate, and define the half-spaces

$$\chi_{j,s,1} = \left\{z \in \mathbb{R}^p : z_j \leq x_{j,(s)}\right\}, \qquad \chi_{j,s,0} = \left\{z \in \mathbb{R}^p : z_j > x_{j,(s)}\right\}. \tag{1}$$

The following definitions are illustrated in Figure 1.

Definition 1 (Tree and Region) *Consider a set $\mathcal{S}$ such that $R \subseteq \mathbb{R}^p$ for all $R \in \mathcal{S}$. Then $\mathcal{S}$ is a* tree *if and only if (i) $\mathbb{R}^p \in \mathcal{S}$; (ii) every element of $\mathcal{S} \setminus \mathbb{R}^p$ equals $R \cap \chi_{j,s,e}$ for some $R \in \mathcal{S}$, $j \in \{1, \ldots, p\}$, $s \in \{1, \ldots, n-1\}$, $e \in \{0, 1\}$; (iii) $R \cap \chi_{j,s,e} \in \mathcal{S}$ implies that $R \cap \chi_{j,s,1-e} \in \mathcal{S}$ for $e \in \{0, 1\}$; and (iv) for any $R, R' \in \mathcal{S}, R \cap R' \in \{\emptyset, R, R'\}$. If $R \in \mathcal{S}$ and $\mathcal{S}$ is a tree, then we refer to R as a* region.

We use the notation TREE to refer to a particular tree. Definition 1 implies that any region $R \in \text{TREE} \setminus \{\mathbb{R}^p\}$ is of the form $R = \cap_{l=1}^{L} \chi_{j_l,s_l,e_l}$, where for each $l = 1, \ldots, L$, we have that $j_l \in \{1, \ldots, p\}$, $s_l \in \{1, \ldots, n-1\}$, and $e_l \in \{0, 1\}$. We call L the *level* of the region, and use the convention that the level of $\mathbb{R}^p$ is 0.

Definition 2 (Siblings and Children) *Suppose that $\{R, R \cap \chi_{j,s,1}, R \cap \chi_{j,s,0}\} \subseteq$ TREE. Then $R \cap \chi_{j,s,1}$ and $R \cap \chi_{j,s,0}$ are* siblings. *Furthermore, they are the* children *of R.*

Definition 3 (Descendant and Ancestor) *If $R, R' \in$ TREE and $R \subseteq R'$, then R is a* descendant *of R', and R' is an* ancestor *of R.*

Definition 4 (Terminal Region) *A region $R \in$ TREE without descendants is a* terminal region.

We let $\text{DESC}(R, \text{TREE})$ denote the set of descendants of region R in TREE, and we let $\text{TERM}(R, \text{TREE})$ denote the subset of $\text{DESC}(R, \text{TREE})$ that are terminal regions.

Given a response vector $y \in \mathbb{R}^n$, let $\bar{y}_R = \left(\sum_{i:x_i \in R} y_i\right) / \left\{\sum_{i=1}^{n} 1_{(x_i \in R)}\right\}$, where $1_{(A)}$ is an indicator variable that equals 1 if the event A holds, and 0 otherwise. Then, a tree TREE induces the regression model $\hat{\mu}(x) = \sum_{R \in \text{TERM}(\mathbb{R}^p, \text{TREE})} \bar{y}_R 1_{(x \in R)}$. In other words, it predicts the response within each terminal region to be the mean of the observations in that region.

2.2 A Review of the CART Algorithm (Breiman et al., 1984)

The CART algorithm (Breiman et al., 1984) greedily searches for a tree that minimizes the sum of squared errors $\sum_{R \in \text{TERM}(\mathbb{R}^p, \text{TREE})} \sum_{i:x_i \in R} (y_i - \bar{y}_R)^2$. It first grows a very large tree via recursive binary splits, starting with the full covariate space $\mathbb{R}^p$. To split a region R, it selects the covariate x_j and the split point $x_{j,(s)}$ to maximize the *gain*, defined as

$$\text{GAIN}_R(y, j, s) \equiv \sum_{i \in R} (y_i - \bar{y}_R)^2 - \left\{ \sum_{i \in R \cap \chi_{j,s,1}} \left(y_i - \bar{y}_{R \cap \chi_{j,s,1}}\right)^2 + \sum_{i \in R \cap \chi_{j,s,0}} \left(y_i - \bar{y}_{R \cap \chi_{j,s,0}}\right)^2 \right\}. \tag{2}$$

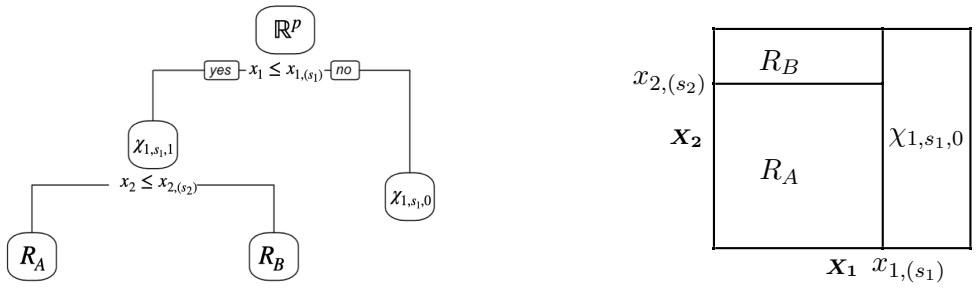

Figure 1: The regression tree takes the form $\text{TREE} = \{\mathbb{R}^p, \chi_{1,s_1,1}, \chi_{1,s_1,0}, \chi_{1,s_1,1} \cap \chi_{2,s_2,1}, \chi_{1,s_1,1} \cap \chi_{2,s_2,0}\}$. The regions $R_A = \chi_{1,s_1,1} \cap \chi_{2,s_2,1}$ and $R_B = \chi_{1,s_1,1} \cap \chi_{2,s_2,0}$ are siblings, and are children, and therefore descendants, of the region $\chi_{1,s_1,1}$. The ancestors of R_A and R_B are $\mathbb{R}^p$ and $\chi_{1,s_1,1}$. Furthermore, R_A, R_B, and $\chi_{1,s_1,0}$ are terminal regions.

Details are provided in Algorithm A1.

Once a very large tree has been grown, cost-complexity pruning is applied. We define the average per-region gain in sum-of-squared errors provided by the descendants of a region R,

$$g(R, \text{TREE}, y) = \frac{\sum_{i:x_i \in R} (y_i - \bar{y}_R)^2 - \sum_{r \in \text{TERM}(R,\text{TREE})} \sum_{i:x_i \in r} (y_i - \bar{y}_r)^2}{|\text{TERM}(R, \text{TREE})| - 1}. \quad (3)$$

Given a complexity parameter $\lambda \ge 0$, if $g(R, \text{TREE}, y) < \lambda$ for some $R \in \text{TREE}$, then cost-complexity pruning removes R's descendants from TREE, turning R into a terminal region. Details are in Algorithm A2, which involves the notion of a *bottom-up ordering*.

Definition 5 (Bottom-up ordering) *Let* $\text{TREE} = \{R_1, \ldots, R_K\}$. *Let* π *be a permutation of the integers* $(1, \ldots, K)$. *Then* $\mathcal{O} = \left(R_{\pi(1)}, \ldots, R_{\pi(K)}\right)$ *is a* bottom-up ordering *of the regions in* TREE *if, for all* $k = 1, \ldots, K$, $\pi(k) \le \pi(j)$ *if* $R_k \in \text{DESC}(R_j, \text{TREE})$.

There are other equivalent formulations for cost-complexity pruning (see Proposition 7.2 in Ripley (1996)); the formulation in Algorithm A2 is convenient for establishing the results in this paper.

To summarize, the CART algorithm first applies Algorithm A1 to the initial region $\mathbb{R}^p$ and the data y to obtain an unpruned tree, which we call $\text{TREE}^0(y)$. It then applies Algorithm A2 to $\text{TREE}^0(y)$ to obtain an optimally-pruned tree using complexity parameter λ, which we call $\text{TREE}^\lambda(y)$.

Algorithm A1 (Growing a tree)

`Grow(`R, y`)`

1. *If a stopping condition is met, return* R.
2. *Else return* $\{R$, `Grow(`$R \cap \chi_{\tilde{j},\tilde{s},1}, y$`)`, `Grow(`$R \cap \chi_{\tilde{j},\tilde{s},0}, y$`)`$\}$, *where*
$$(\tilde{j}, \tilde{s}) \in \arg\max_{(j,s):s \in \{1,\ldots,n-1\}, j \in \{1,\ldots,p\}} \text{GAIN}_R(y, j, s).$$

Algorithm A2 (Cost-complexity pruning) *Parameter $\mathcal{O}$ is a bottom-up ordering of the K regions in* TREE.
PRUNE(TREE, y, λ, $\mathcal{O}$)

1. *Let* $\text{TREE}_0 = \text{TREE}$. *Let K be the number of regions in* TREE_0.
2. *For* $k = 1, \ldots, K$:
   (a) *Let R be the kth region in $\mathcal{O}$.*
   (b) *Update* TREE_k *as follows, where $g(\cdot)$ is defined in* (3):
   $$\text{TREE}_k \leftarrow \begin{cases} \text{TREE}_{k-1} \setminus \text{DESC}(R, \text{TREE}_{k-1}) & \text{if } g(R, \text{TREE}_{k-1}, y) < \lambda, \\ \text{TREE}_{k-1} & \text{otherwise.} \end{cases}$$
3. *Return* TREE_K.

2.3 A Brief Overview of Selective Inference

Here, we provide a very brief overview of selective inference; see Fithian et al. (2014) or Taylor and Tibshirani (2015) for a more detailed treatment.

Consider conducting inference on a parameter θ. Classical approaches assume that we were already interested in conducting inference on θ before looking at our data. If, instead, our interest in θ was sparked by looking at our data, then inference must be performed with care: we must account for the fact that we "selected" θ based on the data (Fithian et al., 2014). In this setting, interest focuses on a p-value $p(Y)$ such that the test for $H_0 : \theta = \theta_0$ based on $p(Y)$ controls the *selective Type 1 error* rate, in the sense that

$$pr_{H_0:\theta=\theta_0} \{p(Y) \leq \alpha \mid \theta \text{ selected}\} \leq \alpha, \text{ for all } 0 \leq \alpha \leq 1. \tag{4}$$

Also of interest are confidence intervals $[L(Y), U(Y)]$ that achieve $(1-\alpha)$-*selective coverage* for the parameter θ, meaning that

$$pr \{\theta \in [L(Y), U(Y)] \mid \theta \text{ selected}\} \geq 1 - \alpha. \tag{5}$$

Roughly speaking, the inferential guarantees in (4) and (5) can be achieved by defining p-values and confidence intervals that condition on the aspect of the data that led to the selection of θ. In recent years, a number of papers have taken this approach to perform selective inference on parameters selected from the data in the regression (Lee et al., 2016; Liu et al., 2018; Tian and Taylor, 2018; Tibshirani et al., 2016), clustering (Gao et al., 2020), and changepoint detection (Hyun et al., 2021; Jewell et al., 2022) settings.

In the next section, we propose p-values that satisfy (4) and confidence intervals that satisfy (5) in the setting of CART, where the parameter of interest is either the mean response within a region, or the difference between the mean responses of two sibling regions.

3. The Selective Inference Framework for CART

3.1 Inference on a Pair of Sibling Regions

Throughout this paper, we assume that $Y \sim N_n(\mu, \sigma^2 I_n)$ with $\sigma > 0$ known.

We let $X \in \mathbb{R}^{n \times p}$ denote a fixed covariate matrix. Suppose that we apply CART with complexity parameter λ to a realization $y = (y_1, \ldots, y_n)^{\text{T}}$ from Y to obtain $\text{TREE}^{\lambda}(y)$. Given

sibling regions R_A and R_B in $\text{TREE}^\lambda(y)$, we define a contrast vector $\nu_{sib} \in \mathbb{R}^n$ such that

$$(\nu_{sib})_i = \frac{1_{(x_i \in R_A)}}{\sum_{i'=1}^n 1_{(x_{i'} \in R_A)}} - \frac{1_{(x_i \in R_B)}}{\sum_{i'=1}^n 1_{(x_{i'} \in R_B)}}, \tag{6}$$

and $\nu_{sib}^{\mathrm{T}}\mu = \left(\sum_{i:x_i \in R_A} \mu_i\right) / \left\{\sum_{i=1}^n 1_{(x_i \in R_A)}\right\} - \left(\sum_{i:x_i \in R_B} \mu_i\right) / \left\{\sum_{i=1}^n 1_{(x_i \in R_B)}\right\}$. Now, consider testing the null hypothesis of no difference in means between R_A and R_B, i.e. $H_0 : \nu_{sib}^{\mathrm{T}}\mu = 0$ versus $H_1 : \nu_{sib}^{\mathrm{T}}\mu \neq 0$. This null hypothesis is of interest because R_A and R_B appeared as siblings in $\text{TREE}^\lambda(y)$. A test based on a p-value of the form $pr_{H_0}\left(|\nu_{sib}^{\mathrm{T}}Y| \geq |\nu_{sib}^{\mathrm{T}}y|\right)$ that does not account for this will not control the selective Type 1 error rate in (4).

To control the selective Type 1 error rate, we propose a p-value that conditions on the aspect of the data that led us to select $\nu_{sib}^{\mathrm{T}}\mu$,

$$pr_{H_0}\left\{|\nu_{sib}^{\mathrm{T}}Y| \geq |\nu_{sib}^{\mathrm{T}}y| \mid R_A, R_B \text{ are siblings in } \text{TREE}^\lambda(Y)\right\}. \tag{7}$$

But (7) depends on a nuisance parameter, the portion of μ that is orthogonal to ν_{sib}. To remove the dependence on this nuisance parameter, we condition on its sufficient statistic $\mathcal{P}^\perp_{\nu_{sib}}Y$, where $\mathcal{P}^\perp_\nu = I - \nu\nu^{\mathrm{T}}/\|\nu\|_2^2$. The resulting p-value, or "tree-value", is defined as

$$p_{sib}(y) = pr_{H_0}\left\{|\nu_{sib}^{\mathrm{T}}Y| \geq |\nu_{sib}^{\mathrm{T}}y| \mid R_A, R_B \text{ are siblings in } \text{TREE}^\lambda(Y), \mathcal{P}^\perp_{\nu_{sib}}Y = \mathcal{P}^\perp_{\nu_{sib}}y\right\}. \tag{8}$$

Results similar to Theorem 6 can be found in Jewell et al. (2022); Lee et al. (2016); Liu et al. (2018), and Tibshirani et al. (2016).

Theorem 6 *The test based on the p-value $p_{sib}(y)$ in (8) controls the selective Type 1 error rate for $H_0 : \nu_{sib}^{\mathrm{T}}\mu = 0$, where ν_{sib} is defined in (6), in the sense that*

$$pr_{H_0}\left\{p_{sib}(Y) \leq \alpha \mid R_A, R_B \text{ are siblings in } \text{TREE}^\lambda(Y)\right\} = \alpha, \text{ for all } 0 \leq \alpha \leq 1. \tag{9}$$

Furthermore, $p_{sib}(y) = pr\left\{|\phi| \geq |\nu_{sib}^{\mathrm{T}}y| \mid \phi \in S^\lambda_{sib}(\nu_{sib})\right\}$, where $\phi \sim N(0, \|\nu_{sib}\|_2^2\sigma^2)$, $y'(\phi, \nu) = \mathcal{P}^\perp_\nu y + \phi(\nu/\|\nu\|_2^2)$, and

$$S^\lambda_{sib}(\nu_{sib}) = \{\phi : R_A, R_B \text{ are siblings in } \text{TREE}^\lambda\{y'(\phi, \nu_{sib})\}\}. \tag{10}$$

Proofs of all theoretical results are provided in the appendix. Theorem 6 says that given the set $S^\lambda_{sib}(\nu_{sib})$, we can compute the p-value in (8) using

$$p_{sib}(y) = 1 - F\left\{|\nu_{sib}^{\mathrm{T}}y|; 0, \|\nu_{sib}\|_2^2\sigma^2, S^\lambda_{sib}(\nu_{sib})\right\} + F\left\{-|\nu_{sib}^{\mathrm{T}}y|; 0, \|\nu_{sib}\|_2^2\sigma^2, S^\lambda_{sib}(\nu_{sib})\right\}, \tag{11}$$

where $F\left(\ \cdot\ ; 0, \|\nu\|^2\sigma^2, S\right)$ denotes the cumulative distribution function of the $N(0, \|\nu\|_2^2\sigma^2)$ distribution truncated to the set S. In Section 4, we provide an efficient approach for analytically characterizing the truncation set $S^\lambda_{sib}(\nu_{sib})$. To avoid numerical issues associated with the truncated normal distribution, we compute (11) using methods described in the supplement of Chen and Bien (2020). Note that the proof of Theorem 6, and consequently the efficient computation of $p_{sib}(y)$ discussed in Section 4, relies on the assumption that $Y \sim N_n(\mu, \sigma^2 I_n)$.

We now consider inverting the test proposed in (8) to construct an equitailed confidence interval for $\nu_{sib}^{\mathrm{T}}\mu$ that has $(1-\alpha)$-selective coverage (5), in the sense that

$$pr\left\{\nu_{sib}^{\mathrm{T}}\mu \in [L(Y), U(Y)] \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y)\right\} = 1-\alpha. \tag{12}$$

Proposition 7 *For any $0 \leq \alpha \leq 1$ and any realization $y \in \mathbb{R}^n$, the values $L(y)$ and $U(y)$ that satisfy*

$$F\{\nu_{sib}^{\mathrm{T}}y; L(y), \sigma^2\|\nu_{sib}\|_2^2, S_{sib}^{\lambda}(\nu_{sib})\} = 1-\alpha/2, \quad F\{\nu_{sib}^{\mathrm{T}}y; U(y), \sigma^2\|\nu_{sib}\|_2^2, S_{sib}^{\lambda}(\nu_{sib})\} = \alpha/2, \tag{13}$$

are unique, and $[L(Y), U(Y)]$ achieves $(1-\alpha)$-selective coverage for $\nu_{sib}^{\mathrm{T}}\mu$.

3.2 Inference on a Single Region

Given a single region R_A in a CART tree, we define the contrast vector ν_{reg} such that

$$(\nu_{reg})_i = 1_{(x_i \in R_A)} \Big/ \left\{\sum_{i'=1}^{n} 1_{(x_{i'} \in R_A)}\right\}. \tag{14}$$

Then, $\nu_{reg}^{\mathrm{T}}\mu = \left(\sum_{i:x_i \in R_A} \mu_i\right) / \left\{\sum_{i=1}^{n} 1_{(x_i \in R_A)}\right\}$. We now consider testing the null hypothesis $H_0 : \nu_{reg}^{\mathrm{T}}\mu = c$ for some fixed c. Because our interest in this null hypothesis results from the fact that $R_A \in \textsc{tree}^{\lambda}(y)$, we must condition on this event in defining the p-value. We define

$$p_{reg}(y) = pr_{H_0}\left\{|\nu_{reg}^{\mathrm{T}}Y - c| \geq |\nu_{reg}^{\mathrm{T}}y - c| \mid R_A \in \textsc{tree}^{\lambda}(Y), \mathcal{P}_{\nu_{reg}}^{\perp}Y = \mathcal{P}_{\nu_{reg}}^{\perp}y\right\}, \tag{15}$$

and introduce the following theorem.

Theorem 8 *The test based on the p-value $p_{reg}(y)$ in (15) controls the selective Type 1 error rate for $H_0 : \nu_{reg}^{\mathrm{T}}\mu = c$, where ν_{reg} is defined in (14). Furthermore, $p_{reg}(y) = pr\left\{|\phi - c| \geq |\nu_{reg}^{\mathrm{T}}y - c| \mid \phi \in S_{reg}(\nu_{reg})\right\}$, where $\phi \sim N(c, \|\nu_{reg}\|_2^2\sigma^2)$ and, for $y'(\phi, \nu) = \mathcal{P}_{\nu}^{\perp}y + \phi(\nu/\|\nu\|_2^2)$,*

$$S_{reg}^{\lambda}(\nu_{reg}) = \{\phi : R_A \in \textsc{tree}^{\lambda}\{y'(\phi, \nu_{reg})\}\}. \tag{16}$$

Theorem 2 and the resulting efficient computations in Section 4 rely on the assumption that $Y \sim N_n(\mu, \sigma^2 I_n)$.

We can also define a confidence interval for $\nu_{reg}^{\mathrm{T}}\mu$ that attains nominal selective coverage.

Proposition 9 *For any $0 \leq \alpha \leq 1$ and any realization $y \in \mathbb{R}^n$, the values $L(y)$ and $U(y)$ that satisfy*

$$F\{\nu_{reg}^{\mathrm{T}}y; L(y), \sigma^2\|\nu_{reg}\|_2^2, S_{reg}^{\lambda}(\nu_{reg})\} = 1-\alpha/2, \quad F\{\nu_{reg}^{\mathrm{T}}y; U(y), \sigma^2\|\nu_{reg}\|_2^2, S_{reg}^{\lambda}(\nu_{reg})\} = \alpha/2, \tag{17}$$

are unique, and $[L(Y), U(Y)]$ achieves $(1-\alpha)$-selective coverage for $\nu_{reg}^{\mathrm{T}}\mu$.

In Section 4, we propose an approach to analytically characterize the set $S_{reg}^{\lambda}(\nu_{reg})$ in (16).

3.3 Intuition for the Conditioning Sets $S^{\lambda}_{sib}(\nu_{sib})$ and $S^{\lambda}_{reg}(\nu_{reg})$

We first develop intuition for the set $S^{\lambda}_{sib}(\nu_{sib})$ defined in (10). From Theorem 6,

$$\{y'(\phi,\nu_{sib})\}_i = y_i + (\phi - \nu_{sib}^{\mathrm{T}} y)\left\{\frac{\sum_{i'=1}^n 1_{(x_{i'}\in R_B)}}{\sum_{i'=1}^n 1_{(x_{i'}\in R_A\cup R_B)}} 1_{(x_i\in R_A)} - \frac{\sum_{i'=1}^n 1_{(x_{i'}\in R_A)}}{\sum_{i'=1}^n 1_{(x_{i'}\in R_A\cup R_B)}} 1_{(x_i\in R_B)}\right\}.$$

Thus, $y'(\phi,\nu_{sib})$ is a perturbation of y that exaggerates the difference between the observed sample mean responses of R_A and R_B if $|\phi| > |\nu_{sib}^{\mathrm{T}} y|$, and shrinks that difference if $|\phi| < |\nu_{sib}^{\mathrm{T}} y|$. The set $S^{\lambda}_{sib}(\nu_{sib})$ quantifies the amount that we can shift the difference in sample mean responses between R_A and R_B while still producing a tree containing these sibling regions. The top row of Figure 2 displays $\textsc{tree}^0\{y'(\phi,\nu_{sib})\}$, as a function of ϕ, in an example where $S^0_{sib}(\nu_{sib}) = (-19.8,-1.8)\cup(0.9,34.9)$.

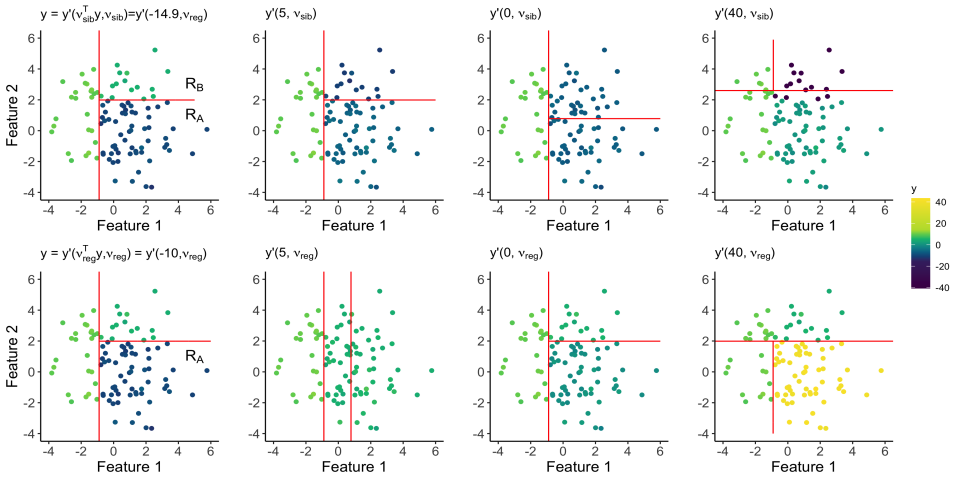

Figure 2: Data with $n = 100$ and $p = 2$. Regions resulting from CART ($\lambda = 0$) are delineated using solid lines. Here, $R_A = \chi_{1,26,0} \cap \chi_{2,72,1}$ and $R_B = \chi_{1,26,0} \cap \chi_{2,72,0}$. *Top:* Output of CART applied to $y'(\phi,\nu_{sib})$, where ν_{sib} in (6) encodes the contrast between R_A and R_B, for various values of ϕ. The left-most panel displays $y = y'(\nu_{sib}^{\mathrm{T}} y, \nu_{sib})$. By inspection, we see that $-14.9 \in S^0_{sib}(\nu_{sib})$ and $5 \in S^0_{sib}(\nu_{sib})$, but $0 \notin S^0_{sib}(\nu_{sib})$ and $40 \notin S^0_{sib}(\nu_{sib})$. In fact, $S^0_{sib}(\nu_{sib}) = (-19.8,-1.8)\cup(0.9,34.9)$. *Bottom:* Output of CART applied to $y'(\phi,\nu_{reg})$, where ν_{reg} in (14) encodes membership in R_A. The left-most panel displays $y = y'(\nu_{reg}^{\mathrm{T}} y, \nu_{reg})$. Here, $S^0_{reg}(\nu_{reg}) = (-\infty,3.1)\cup(5.8,8.8)\cup(14.1,\infty)$.

We next develop intuition for $S^{\lambda}_{reg}(\nu_{reg})$, defined in (16). Note that $\{y'(\phi,\nu_{reg})\}_i = y_i + (\phi - \nu_{reg}^{\mathrm{T}} y)1_{(x_i\in R_A)}$, where $y'(\phi,\nu_{reg})$ is defined in Theorem 8. Thus, $y'(\phi,\nu_{reg})$ shifts the responses of the observations in R_A so that their sample mean equals ϕ, and leaves the others unchanged. The set $S^{\lambda}_{reg}(\nu_{reg})$ quantifies the amount that we can exaggerate or shrink the sample mean response in region R_A while still producing a tree that contains R_A. The bottom row of Figure 2 displays $y'(\phi,\nu_{reg})$ as ϕ is varied, in an example with $S^0_{reg}(\nu_{reg}) = (-\infty,3.1)\cup(5.8,8.8)\cup(14.1,\infty)$.

4. Computing the conditioning sets $S^{\lambda}_{sib}(\nu_{sib})$ and $S^{\lambda}_{reg}(\nu_{reg})$

4.1 Recharacterizing the conditioning sets in terms of branches

We begin by introducing the concept of a branch.

Definition 10 (Branch) *A* branch *is an ordered sequence of triples* $\mathcal{B} = ((j_1, s_1, e_1), \ldots, (j_L, s_L, e_L))$ *such that* $j_l \in \{1, \ldots, p\}$, $s_l \in \{1, \ldots, n-1\}$, *and* $e_l \in \{0, 1\}$ *for* $l = 1, \ldots, L$. *The branch* $\mathcal{B}$ *induces a nested set of regions* $\mathcal{R}(\mathcal{B}) = \{R^{(0)}, R^{(1)}, \ldots, R^{(L)}\}$, *where* $R^{(l)} = \bigcap_{l'=1}^{l} \chi_{j_{l'}, s_{l'}, e_{l'}}$ *for* $l = 1, \ldots, L$, *and* $R^{(0)} = \mathbb{R}^p$.

For a branch $\mathcal{B}$ and a vector ν, we define

$$S^{\lambda}(\mathcal{B}, \nu) = \left\{ \phi : \mathcal{R}(\mathcal{B}) \subseteq \textsc{tree}^{\lambda}\{y'(\phi, \nu)\} \right\}. \tag{18}$$

For $R \in \textsc{tree}$, we let $\textsc{branch}(R, \textsc{tree})$ denote the branch such that $\mathcal{R}\{\textsc{branch}(R, \textsc{tree})\}$ contains R and all of its ancestors in $\textsc{tree}$.

Lemma 11 *Suppose that* R_A *and* R_B *are siblings in* $\textsc{tree}^{\lambda}(y)$. *Then* R_A *and* R_B *are siblings in* $\textsc{tree}^{\lambda}\{y'(\phi, \nu_{sib})\}$ *if and only if* $\mathcal{R}[\textsc{branch}\{R_A, \textsc{tree}^{\lambda}(y)\}] \subseteq \textsc{tree}^{\lambda}\{y'(\phi, \nu_{sib})\}$. *Therefore,* $S^{\lambda}_{sib}(\nu_{sib}) = S^{\lambda}[\textsc{branch}\{R_A, \textsc{tree}^{\lambda}(y)\}, \nu_{sib}]$, *defined in* (10) *and* (18).

Lemma 11 says that $\textsc{tree}^{\lambda}\{y'(\phi, \nu_{sib})\}$ contains siblings R_A and R_B if and only if it contains the entire branch associated with R_A in $\textsc{tree}^{\lambda}(y)$. However, Lemma 11 does not apply in the single region case: for ν_{reg} defined in (14) and some $R_A \in \textsc{tree}^{\lambda}(y)$, the fact that $R_A \in \textsc{tree}^{\lambda}\{y'(\phi, \nu_{reg})\}$ does not imply that $\mathcal{R}[\textsc{branch}\{R_A, \textsc{tree}^{\lambda}(y)\}] \subseteq \textsc{tree}^{\lambda}\{y'(\phi, \nu_{reg})\}$. Instead, a result similar to Lemma 11 holds, involving permutations of the branch.

Definition 12 (Permutation of a branch) *Let* Π *denote the set of all* $L!$ *permutations of* $(1, 2, \ldots, L)$. *Given* $\pi \in \Pi$ *and a branch* $\mathcal{B} = ((j_1, s_1, e_1), \ldots, (j_L, s_L, e_L))$, *we say that* $\pi(\mathcal{B}) = ((j_{\pi(1)}, s_{\pi(1)}, e_{\pi(1)}), \ldots, (j_{\pi(L)}, s_{\pi(L)}, e_{\pi(L)}))$ *is a* permutation of the branch $\mathcal{B}$.

Branch $\mathcal{B}$ and its permutation $\pi(\mathcal{B})$ induce the same region $R^{(L)}$, but $\mathcal{R}\{\pi(\mathcal{B})\} \neq \mathcal{R}(\mathcal{B})$.

Lemma 13 *Let* $R_A \in \textsc{tree}^{\lambda}(y)$. *Then* $R_A \in \textsc{tree}^{\lambda}\{y'(\phi, \nu_{reg})\}$ *if and only if there exists a* $\pi \in \Pi$ *such that* $\mathcal{R}[\pi\{\textsc{branch}_{R_A}(y)\}] \subseteq \textsc{tree}^{\lambda}\{y'(\phi, \nu_{reg})\}$. *Thus, for* $S^{\lambda}_{reg}(\nu_{reg})$ *in* (16),

$$S^{\lambda}_{reg}(\nu_{reg}) = \bigcup_{\pi \in \Pi} S^{\lambda}\left(\pi\left[\textsc{branch}\{R_A, \textsc{tree}^{\lambda}(y)\}\right], \nu_{reg}\right). \tag{19}$$

Lemmas 11 and 13 reveal that computing $S^{\lambda}_{sib}(\nu_{sib})$ and $S^{\lambda}_{reg}(\nu_{reg})$ requires characterizing sets of the form $S^{\lambda}(\mathcal{B}, \nu)$, defined in (18). To compute $S^{\lambda}_{sib}(\nu_{sib})$ we will only need to consider $S^{\lambda}(\mathcal{B}, \nu)$ where $\mathcal{R}(\mathcal{B}) \subseteq \textsc{tree}^{\lambda}(y)$. However, to compute $S^{\lambda}_{reg}(\nu_{reg})$, we will need to consider $S^{\lambda}\{\pi(\mathcal{B}), \nu\}$ where $\mathcal{R}(\mathcal{B}) \subseteq \textsc{tree}^{\lambda}(y)$ but $\mathcal{R}\{\pi(\mathcal{B})\} \nsubseteq \textsc{tree}^{\lambda}(y)$.

4.2 Computing $S^\lambda(\mathcal{B}, \nu)$ in (18)

Throughout this section, we consider a vector $\nu \in \mathbb{R}^n$ and a branch $\mathcal{B} = ((j_1, s_1, e_1), \ldots, (j_L, s_L, e_L))$, where $\mathcal{R}(\mathcal{B})$ may or may not be in $\text{TREE}^\lambda(y)$. Recall from Definition 10 that $\mathcal{B}$ induces the nested regions $R^{(l)} = \bigcap_{l'=1}^{l} \chi_{j_{l'}, s_{l'}, e_{l'}}$ for $l = 1, \ldots, L$, and $R^{(0)} = \mathbb{R}^p$. Throughout this section, our only requirement on $\mathcal{B}$ and ν is the following condition.

Condition 1 *For $y'(\phi, \nu)$ defined in Theorem 6, $\mathcal{B}$ and ν satisfy $\{y'(\phi, \nu)\}_i = y_i + c_1 1_{\{x_i \in R^{(L)}\}} + c_2 1_{[x_i \in \{R^{(L-1)} \cap \chi_{j_L, s_L, 1-e_L}\}]}$ for $i = 1, \ldots, n$ and for some constants c_1 and c_2.*

To characterize $S^\lambda(\mathcal{B}, \nu)$ in (18), recall that the CART algorithm in Section 2.2 involves growing a very large tree $\text{TREE}^0(y)$, and then pruning it. We first characterize the set

$$S_{grow}(\mathcal{B}, \nu) = \{\phi : \mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^0\{y'(\phi, \nu)\}\}. \tag{20}$$

Proposition 14 *Recall the definition of $\text{GAIN}_{R^{(l)}}\{y'(\phi, \nu), j, s\}$ in (2), and let $S_{l,j,s} = \{\phi : \text{GAIN}_{R^{(l-1)}}\{y'(\phi, \nu), j, s\} \leq \text{GAIN}_{R^{(l-1)}}\{y'(\phi, \nu), j_l, s_l\}\}$. Then, $S_{grow}(\mathcal{B}, \nu) = \bigcap_{l=1}^{L} \bigcap_{j=1}^{p} \bigcap_{s=1}^{n-1} S_{l,j,s}$.*

Proposition 15 says that we can compute $S_{grow}(\mathcal{B}, \nu)$ efficiently.

Proposition 15 *The set $S_{l,j,s}$ is defined by a quadratic inequality in ϕ. Furthermore, we can evaluate all of the sets $S_{l,j,s}$, for $l = 1, \ldots, L$, $j = 1, \ldots, p$, $s = 1, \ldots, n-1$, in $O\{npL + np\log(n)\}$ operations. Intersecting these sets to obtain $S_{grow}(\mathcal{B}, \nu)$ requires at most $O\{npL \times \log(npL)\}$ operations, and only $O(npL)$ operations if $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}$ and ν is of the form ν_{sib} in (6).*

Noting that $S^\lambda(\mathcal{B}, \nu) = \left\{\phi \in S_{grow}(\mathcal{B}, \nu) : R^{(L)} \in \text{TREE}^\lambda\{y'(\phi, \nu)\}\right\}$, it remains to characterize the set of $\phi \in S_{grow}(\mathcal{B}, \nu)$ such that $R^{(L)}$ is not removed during pruning. Recall that $g(\cdot)$ was defined in (3).

Proposition 16 *There exists a tree $\text{TREE}(\mathcal{B}, \nu, \lambda)$ such that*

$$S^\lambda(\mathcal{B}, \nu) = S_{grow}(\mathcal{B}, \nu) \cap \left(\bigcap_{l=0}^{L-1} \left\{\phi : g\left\{R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu)\right\} \geq \lambda\right\} \right). \tag{21}$$

If $\mathcal{R}(\mathcal{B}) \in \text{TREE}^\lambda(y)$, then $\text{TREE}(\mathcal{B}, \nu, \lambda) = \text{TREE}^\lambda(y)$ satisfies (21). Otherwise, given the set $S_{grow}(\mathcal{B}, \nu)$, computing a $\text{TREE}(\mathcal{B}, \nu, \lambda)$ that satisfies (21) has a worst-case computational cost of $O(n^2 p)$.

We explain how to compute a $\text{TREE}(\mathcal{B}, \nu, \lambda)$ satisfying (21) when $\mathcal{R}(\mathcal{B}) \notin \text{TREE}^\lambda(y)$ in the supplementary materials.

Proposition 17 *The set $\bigcap_{l=0}^{L-1} \left\{\phi : g\left\{R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu)\right\} \geq \lambda\right\}$ in (21) is the intersection of the solution sets of L quadratic inequalities in ϕ. Given $\text{TREE}(\mathcal{B}, \nu, \lambda)$, the coefficients of these quadratics can be obtained in $O(nL)$ operations. After $S_{grow}(\mathcal{B}, \nu)$ has been computed, intersecting it with these quadratic sets to obtain $S^\lambda(\mathcal{B}, \nu)$ from (21) requires $O\{npL \times \log(npL)\}$ operations in general, and only $O(L)$ operations if $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}$ and $\nu = \nu_{sib}$ from (6).*

The results in this section have relied upon Condition 1. Indeed, this condition holds for branches $\mathcal{B}$ and vectors ν that arise in characterizing the sets $S^{\lambda}_{sib}(\nu_{sib})$ and $S^{\lambda}_{reg}(\nu_{reg})$.

Proposition 18 *If either (i)* $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$ *and* $\nu = \nu_{sib}$ (6)*, where* R_A *and* R_B *are siblings in* $\text{TREE}^{\lambda}(y)$*, or (ii)* $\mathcal{B}$ *is a permutation of* $\text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$ *and* $\nu = \nu_{reg}$ (14)*, where* $R_A \in \text{TREE}^{\lambda}(y)$*, then Condition 1 holds.*

Combining Lemma 11 with Propositions 14–18, we see that $S^{\lambda}_{sib}(\nu_{sib})$ can be computed in $O\{npL + np\log(n)\}$ operations. However, computing $S^{\lambda}_{reg}(\nu_{reg})$ is much more computationally intensive: by Lemma 13 and Propositions 14–18, it requires computing $S^{\lambda}(\pi\left[\text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}\right], \nu_{reg})$ for all $L!$ permutations $\pi \in \Pi$, for a total of $O\left[L!\left\{n^2 pLlog(pL)\right\}\right]$ operations. In Section 4.3, we discuss ways to avoid these calculations.

4.3 A Computationally-Efficient Alternative to $S^{\lambda}_{reg}(\nu_{reg})$

Lemma 13 suggests that carrying out inference on a single region requires computing $S^{\lambda}(\pi\left[\text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}\right], \nu_{reg})$ for every $\pi \in \Pi$. We now present a less computationally demanding alternative.

Proposition 19 *Let* Q *be a subset of the* $L!$ *permutations in* Π*, i.e.* $Q \subseteq \Pi$*. Define*

$$p^{Q}_{reg}(y) = pr_{H_0}\left\{|\nu^{\text{T}}_{reg}Y - c| \geq |\nu^{\text{T}}_{reg}y - c| \mid \bigcup_{\pi \in Q}\left(\mathcal{R}(\pi[\text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}]) \subseteq \text{TREE}^{\lambda}(Y)\right), \mathcal{P}^{\perp}_{\nu_{reg}}Y = \mathcal{P}^{\perp}_{\nu_{reg}}y\right\}.$$

The test based on $p^{Q}_{reg}(y)$ *controls the selective Type 1 error rate* (4) *for* $H_0 : \nu^{\text{T}}_{reg}\mu = c$*. Furthermore,* $p^{Q}_{reg}(y) = pr\left\{|\phi - c| \geq |\nu^{\text{T}}_{reg}y - c| \mid \phi \in \bigcup_{\pi \in Q} S^{\lambda}\left(\pi[\text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}], \nu_{reg}\right)\right\}$*, where* $\phi \sim N(c, \|\nu_{reg}\|^2_2\sigma^2)$*.*

Using the notation in Proposition 19, $p_{reg}(y)$ introduced in (15) equals $p^{\Pi}_{reg}(y)$. If we take $Q = \{\mathcal{I}\}$, where $\mathcal{I}$ is the identity permutation, then we arrive at

$$p^{\mathcal{I}}_{reg}(y) = P\left(|\phi - c| \geq |\nu^{\text{T}}_{reg}y - c| \mid \phi \in S^{\lambda}[\text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}, \nu_{reg}]\right), \qquad (22)$$

where $\phi \sim N(c, \|\nu_{reg}\|^2_2\sigma^2)$. The set $S^{\lambda}[\text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}, \nu_{reg}]$ can be easily computed by Proposition 16.

Compared to (15), (22) conditions on an extra piece of information: the ancestors of R_A. Thus, while (22) controls the selective Type 1 error rate, it may have lower power than (15) (Fithian et al., 2014). Similarly, inverting (22) to form a confidence interval provides correct selective coverage, but may yield intervals that are wider than those in Proposition 9. Proposition 19 is motivated by a proposal by Lee et al. (2016) to condition on both the selected model (necessary information) and the signs of the selected variables (extra information) in the lasso setting, to gain computational efficiency at the possible expense of precision and power.

In Appendix F, we show through simulation that the loss in power associated with using (22) rather than (15) is negligible. Thus, in practice, we suggest using (22) for its computational efficiency. We use (22) for the remainder of this paper.

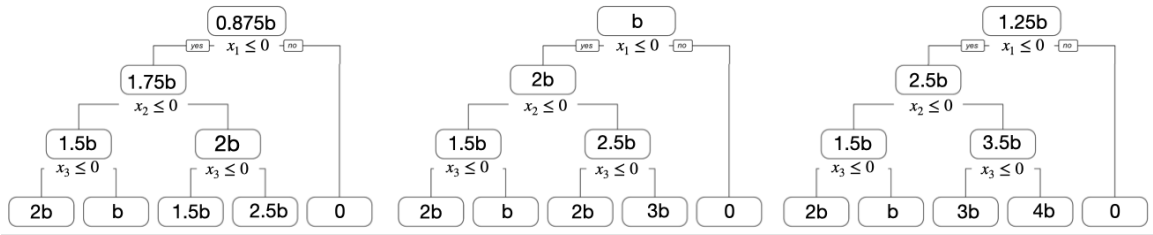

Figure 3: The true mean model in Section 5, for $a = 0.5$ (left), $a = 1$ (center), and $a = 2$ (right). The difference in means between the sibling nodes at level two in the tree is ab, while the difference in means between the sibling nodes at level three is b.

Furthermore, we can consider computing confidence intervals of the form $[L_{S^{\mathcal{I}}_{reg}}(y), U_{S^{\mathcal{I}}_{reg}}(y)]$ rather than (17), where $L_{S^{\mathcal{I}}_{reg}}(y)$ and $U_{S^{\mathcal{I}}_{reg}}(y)$ satisfy

$$F\left(\nu_{reg}^{\mathrm{T}} y; L_{S^{\mathcal{I}}_{reg}}(y), \sigma^2 \|\nu_{reg}\|_2^2, S^{\lambda}\left[\textsc{branch}\{R_A, \textsc{tree}^{\lambda}(y)\}, \nu_{reg}\right]\right) = 1 - \frac{\alpha}{2},$$
$$F\left(\nu_{reg}^{\mathrm{T}} y; U_{S^{\mathcal{I}}_{reg}}(y), \sigma^2 \|\nu_{reg}\|_2^2, S^{\lambda}\left[\textsc{branch}\{R_A, \textsc{tree}^{\lambda}(y)\}, \nu_{reg}\right]\right) = \frac{\alpha}{2}. \tag{23}$$

In Appendix F, we show that the confidence intervals resulting from (23) are not much wider than those resulting from (17). We therefore make use of confidence intervals of the form (23) in the remainder of this paper.

5. Simulation Study

5.1 Data Generating Mechanism

We simulate $X \in \mathbb{R}^{n \times p}$ with $n = 200, p = 10$, $X_{ij} \overset{i.i.d.}{\sim} N(0,1)$, and $y \sim N_n(\mu, \sigma^2 I_n)$ with $\sigma = 5$ and $\mu_i = b \times \left[1_{(x_{i,1} \leq 0)} \times \{1 + a 1_{(x_{i,2} > 0)} + 1_{(x_{i,3} \times x_{i,2} > 0)}\}\right]$. This μ vector defines a three-level tree, shown in Figure 3 for three values of $a \in \mathbb{R}$.

5.2 Methods for Comparison

All CART trees are fit using the `R` package `rpart` (Therneau and Atkinson, 2019) with $\lambda = 200$, a maximum level of three, and a minimum node size of one. We compare three approaches for conducting inference. (i) *Selective Z-methods:* Fit a CART tree to the data. For each split, test for a difference in means between the two sibling regions using (8), and compute the corresponding confidence interval in (13). Compute the confidence interval for the mean of each region using (23). (ii) *Naive Z-methods:* Fit a CART tree to the data. For each split, conduct a naive Z-test for the difference in means between the two sibling regions, and compute the corresponding naive Z-interval. Compute a naive Z-interval for each region's mean. (iii) *Sample splitting:* Split the data into equally-sized training and test sets. Fit a CART tree to the training set. On the test set, conduct a naive Z-test for each

split and compute a naive Z-interval for each split and each region. If a region has no test set observations, then we fail to reject the null hypothesis and fail to cover the parameter.

The conditional inference tree (CTree) framework of Hothorn et al. (2006) uses a different criterion than CART to perform binary splits. Within a region, it tests for linear association between each covariate and the response. The covariate with the smallest p-value for this linear association is selected as the split variable, and a Bonferroni corrected p-value that accounts for the number of covariates is reported in the final tree. Then, the split point is selected. If, after accounting for multiple testing, no variable has a p-value below a pre-specified significance level α, then the recursion stops. While CTree's p-values assess linear association and thus are not directly comparable to the p-values in (i)–(iii) above, it is the most popular framework currently available for determining if a regression tree split is statistically significant. Thus, we also evaluate the performance of (iv) *CTree:* Fit a CTree to all of the data using the `R` package `partykit` (Hothorn and Zeileis, 2015) with $\alpha = 0.05$. For each split, record the p-value reported by `partykit`.

In Sections 5.3–5.6, we assume that σ is known. We consider the case of unknown σ in Section 5.7.

5.3 Uniform p-values under a Global Null

We generate $5,000$ datasets with $a = b = 0$, so that $H_0 : \nu_{sib}^{\mathrm{T}}\mu = 0$ holds for all splits in all trees. Figure 4 displays the distributions of p-values across all splits in all fitted trees for the naive Z-test, sample splitting, and the selective Z-test. The selective Z-test and sample splitting achieve uniform p-values under the null, while the naive Z-test (which does not account for the fact that ν_{sib} was obtained by applying CART to the same data used for testing) does not. CTree is omitted from the comparison: it creates a split only if the p-value is less than $\alpha = 0.05$, and thus its p-values over the splits do not follow a Uniform(0,1) distribution.

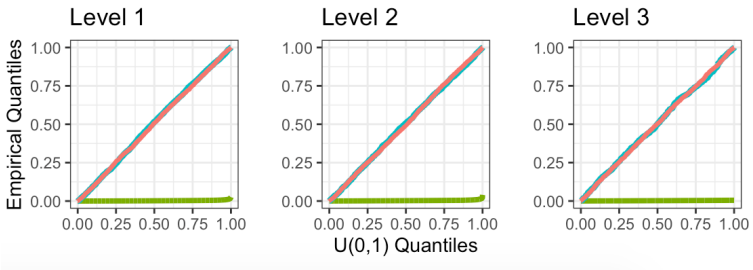

Figure 4: Quantile-quantile plots of the p-values for testing $H_0 : \nu_{sib}^{\mathrm{T}}\mu = 0$, as described in Section 5.3. A naive Z-test (green), sample splitting (blue), and selective Z-test (pink) were performed; see Section 5.2. The p-values are stratified by the level of the regions in the fitted tree.

Table 1: A 3×3 contingency table indicating an observation's involvement in a given true split and estimated split. The adjusted Rand index is computed using only the shaded cells

		Estimated Split		
		In left region	In right region	In neither
True Split	In left region	t_1	t_2	t_3
	In right region	u_1	u_2	u_3
	In neither	v_1	v_2	v_3

5.4 Power

We generate 500 datasets for each $(a, b) \in \{0.5, 1, 2\} \times \{1, \ldots, 10\}$, and evaluate the power of selective Z-tests, sample splitting, and CTree to reject the null hypothesis $H_0 : \nu_{sib}^{\mathrm{T}} \mu = 0$. As naive Z-tests do not control the Type 1 error rate (Figure 4), we do not evaluate their power. We consider two aspects of power: the probability that we *detect* a true split, and the probability that we *reject* the null hypothesis corresponding to a true split.

Given a true split in Figure 3 and an estimated split, we construct the 3×3 contingency table in Table 1, which indicates whether an observation is on the left-hand side, right-hand side, or not involved in the true split (rows) and the estimated split (columns). To quantify the agreement between the true and estimated splits, we compute the adjusted Rand index (Hubert and Arabie, 1985) associated with the 2×3 contingency table corresponding to the shaded region in Table 1. For each true split, we identify the estimated split for which the adjusted Rand index is largest; if this index exceeds 0.75 then this true split is "detected". Given that a true split is detected, the associated null hypothesis is rejected if the corresponding p-value is below 0.05. Figure 5 displays the proportion of true splits that are detected and rejected by each method.

As sample splitting fits a tree using only half of the data, it detects fewer true splits, and thus rejects the null hypothesis for fewer true splits, than the selective Z-test.

When a is small, the difference in means between sibling regions at level two is small. Because CTree makes a split only if there is strong evidence of association at that level, it tends to build one-level trees, and thus fails to detect many true splits; by contrast, the selective Z-test (based on CART) successfully builds more three-level trees. Thus, when a is small, the selective Z-test detects (and rejects) more true differences than CTree between regions at levels two and three.

5.5 Coverage of Confidence Intervals for $\nu_{sib}^{\mathrm{T}} \mu$ and $\nu_{reg}^{\mathrm{T}} \mu$

We generate 500 datasets for each $(a, b) \in \{0.5, 1, 2\} \times \{0, \ldots, 10\}$ to evaluate the coverage of 95% confidence intervals constructed using naive Z-methods, selective Z-methods, and sample splitting. CTree is omitted from these comparisons because it does not provide confidence intervals. We say that the interval covers the truth if it contains $\nu^{\mathrm{T}} \mu$, where ν is defined as in (6) (for a particular split) or (14) (for a particular region). Table 2 shows the proportion of each type of interval that covers the truth, aggregated across values of a

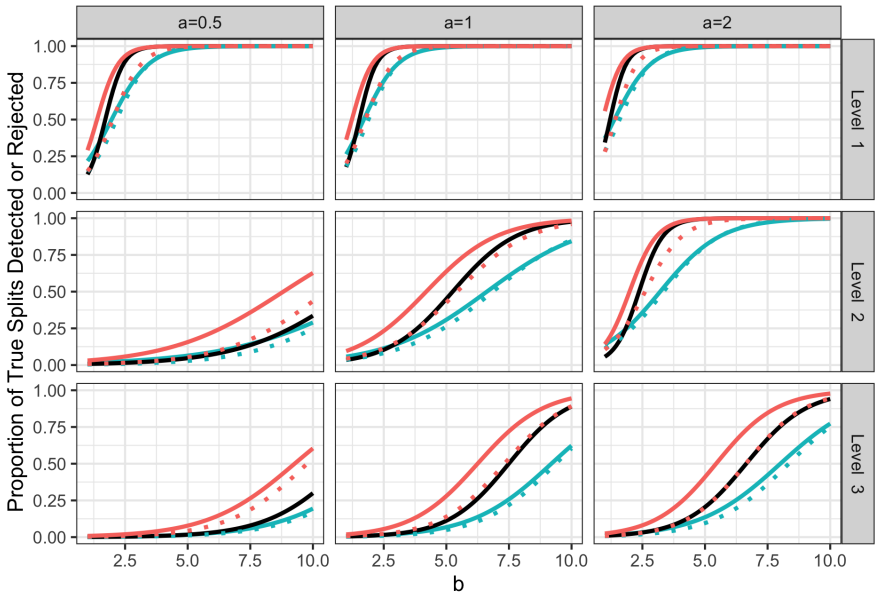

Figure 5: Proportion of true splits detected (solid lines) and rejected (dotted lines) for CART with selective Z-tests (pink), CTree (black), and CART with sample splitting (blue) across different settings of the data generating mechanism, stratified by level in tree. As CTree only makes a split if the p-value is less than 0.05, the proportion of detections equals the proportion of rejections.

and b. The selective Z-intervals attain correct coverage of 95%, while the naive Z-intervals do not.

It may come as a surprise that sample splitting does not attain correct coverage. Recall that ν from (6) or (14) is an n-vector that contains entries for all observations in both the training set and the test set. Thus, $\nu^{\mathrm{T}}\mu$ involves the true mean among both training and test set observations in a given region or pair of regions. By contrast, sample splitting attains correct coverage for a different parameter involving the true means of only the test observations that fall within a given region or pair of regions.

5.6 Width of Confidence Intervals

Figure 6(a) illustrates that our selective Z-intervals for $\nu_{reg}^{\mathrm{T}}\mu$ can be extremely wide when b is small, particularly for regions located at deeper levels in the tree. For each tree that we build and for levels 1, 2, and 3, we compute the adjusted Rand Index (Hubert and Arabie, 1985) between the true tree (truncated at the appropriate level) and the estimated tree (truncated at the same level). Figure 6(b) shows that our selective confidence intervals can be extremely wide when this adjusted Rand Index is small, particularly at deeper levels of the tree.

When b is small and the adjusted Rand Index is small, the trees built by CART tend to be unstable, in the sense that small perturbations to the data affect the fitted tree. In this

Table 2: Proportion of 95% confidence intervals containing the true parameter, aggregated over all trees fit to the 5,500 datasets generated with $(a, b) \in \{0.5, 1, 2\} \times \{1, \ldots, 10\}$

	Parameter $\nu_{reg}^{\mathrm{T}}\mu$			Parameter $\nu_{sib}^{\mathrm{T}}\mu$		
Level	Selective Z	Naive Z	Sample Splitting	Selective Z	Naive Z	Sample Splitting
1	0.951	0.889	0.918	0.948	0.834	0.915
2	0.950	0.645	0.921	0.951	0.410	0.917
3	0.951	0.711	0.921	0.950	0.550	0.921

setting, the sample statistics $\nu_{reg}^{\mathrm{T}}y$ fall very close to the boundary of the truncation set. See Kivaranovic and Leeb (2021) for a discussion of why wide confidence intervals can arise in these settings. The great width of our confidence intervals reflects the uncertainty about the mean response within each region due to the instability of the tree-fitting procedure.

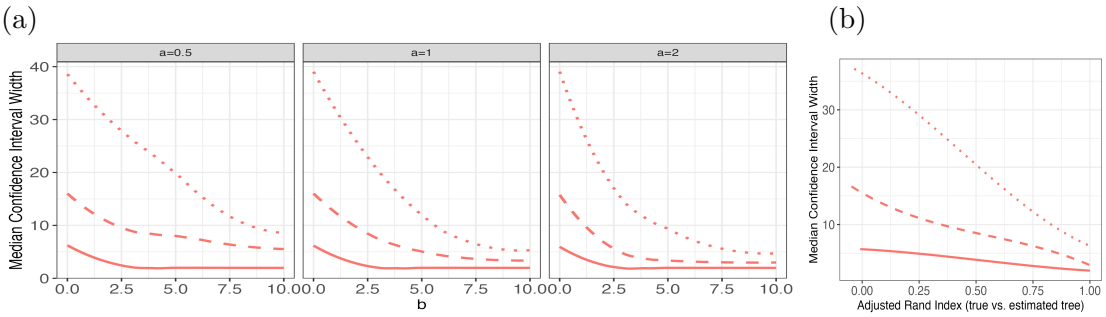

Figure 6: The median width of the selective Z-intervals for parameter $\nu_{reg}^{\mathrm{T}}\mu$ for regions at levels one (solid), two (dashed), and three (dotted) of the tree. Similar results hold for parameter $\nu_{sib}^{\mathrm{T}}\mu$. Panel (a) breaks results down by the parameters a and b, whereas panel (b) aggregates results across values of parameters a and b, and displays them as a function of the adjusted Rand Index between the true and estimated trees.

5.7 Results with Unknown σ

Thus far, we have assumed that σ is known. In this section, we compare the following three versions of the selective Z-methods that plug different values of σ into the truncated normal CDF when computing p-values and confidence intervals:

1. σ: We plug in the true value of σ, as in Sections 5.3–5.6.

2. $\hat{\sigma}_{\text{cons}}$: We plug in $\hat{\sigma}_{\text{cons}} = \sqrt{(n-1)^{-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$, where $\bar{y} = n^{-1}\sum_{i=1}^{n} y_i$.

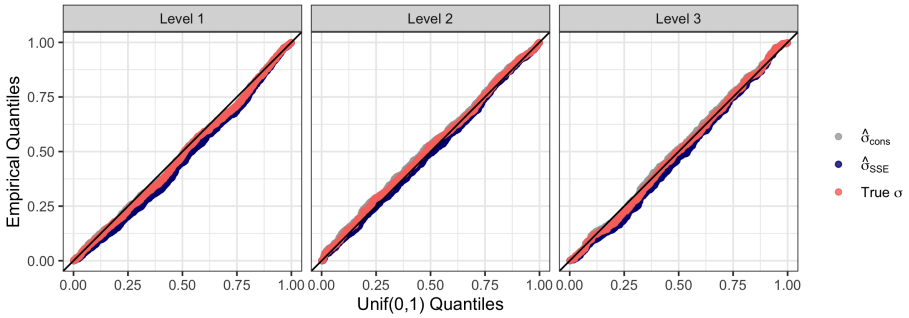

Figure 7: QQ plots of the p-values from testing $H_0 : \nu_{sib}^T \mu = 0$ when $\mu = 0_n$ using the selective Z-test with three different values plugged in to the truncated normal CDF for σ. The p-values are stratified by the level of the regions in the fitted tree.

3. $\hat{\sigma}_{\text{SSE}}$: Let $\mathcal{T} = \left|\text{TERM}\left(\mathbb{R}^p, \text{TREE}^\lambda(y)\right)\right|$ be the number of terminal regions in $\text{TREE}^\lambda(y)$. We plug in $\hat{\sigma}_{\text{SSE}} = \sqrt{(n-\mathcal{T})^{-1}\sum_{i=1}^n (y_i - \hat{y}_i)^2}$, where $\hat{y}_i$ is the predicted value for the ith observation given by $\text{TREE}^\lambda(y)$.

It is straightforward to show that $E[\hat{\sigma}^2_{\text{cons}}] \geq \sigma^2$, for any value of $E[y] = \mu$. Thus, we expect this estimate to lead to conservative inference. On the other hand, $\hat{\sigma}^2_{\text{SSE}}$ can be made arbitrarily small by making the fitted tree arbitrarily deep, and so we expect inference based on this estimate to be anti-conservative if the fitted CART tree is large.

Figure 7 shows the distribution of p-values from testing $H_0 : \nu_{sib}^T \mu = 0$ with the three versions of the selective Z-test under the data generating mechanism described in Section 5.1, with $a = b = 0$. In this setting, $\nu_{sib}^T \mu = 0$ holds for all splits in all trees. We see almost no difference between the three versions of the selective Z-test. In this global null setting, $E[\hat{\sigma}^2_{\text{cons}}] = \sigma^2$. Furthermore, the empirical bias of $\hat{\sigma}^2_{\text{SSE}}$ is small because the trees we grow are not particularly large; as in the rest of Section 5, we build trees to a maximum depth of 3 and prune with $\lambda = 200$.

Figure 8 displays the proportion of true splits detected and the proportion of true splits detected and rejected, as defined in Section 5.4, for the three versions of the selective Z-test when data is generated as in Section 5.4. For simplicity, we only show the setting where $a = 1$. All three methods detect the same proportion of true splits, because they all perform inference on the same CART trees. The proportion of splits detected and rejected is very similar for σ and $\hat{\sigma}_{\text{SSE}}$ because $\hat{\sigma}_{\text{SSE}}$ is a very good estimator for σ in this setting. While $\hat{\sigma}_{\text{cons}}$ performs reasonably when b is small, it severely overestimates σ and thus has low power when b is large.

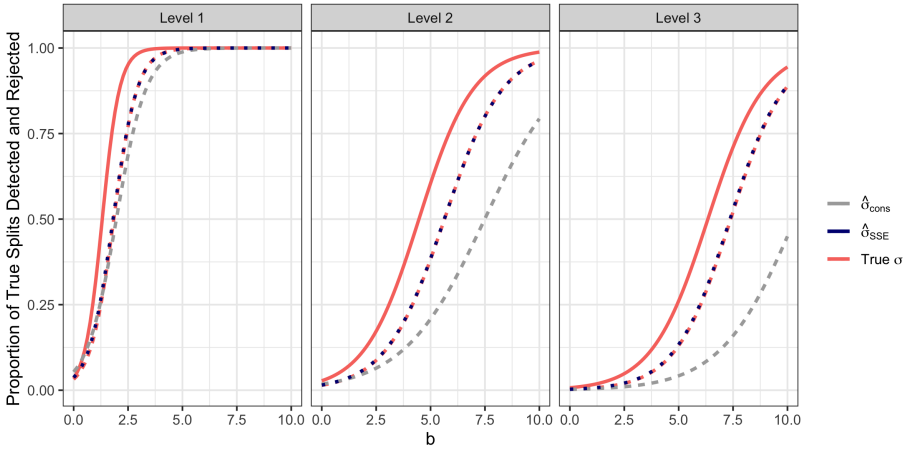

Figure 8: Proportion of true splits detected (solid lines) and rejected (dotted lines) for CART with the three versions of the selective Z-test. The results are stratified by level in tree.

Table 3 displays confidence intervals for $\nu_{sib}^{\mathrm{T}}\mu$ and $\nu_{reg}^{\mathrm{T}}\mu$ for the three versions of the selective Z-intervals, where data is generated as in Section 5.5. As expected, $\hat{\sigma}_{\text{cons}}$ leads to slight over-coverage and $\hat{\sigma}_{\text{SSE}}$ leads to slight under-coverage.

	Parameter $\nu_{reg}^{\mathrm{T}}\mu$			Parameter $\nu_{sib}^{\mathrm{T}}\mu$		
Level	σ	$\hat{\sigma}_{\text{cons}}$	$\hat{\sigma}_{\text{SSE}}$	σ	$\hat{\sigma}_{\text{cons}}$	$\hat{\sigma}_{\text{SSE}}$
1	0.95	0.98	0.95	0.95	0.98	0.94
2	0.95	0.97	0.94	0.95	0.97	0.94
3	0.95	0.96	0.94	0.95	0.96	0.94

Table 3: Proportion of 95% confidence intervals containing the true parameter, aggregated over all trees fit to the 5,500 datasets generated with $(a, b) \in \{0.5, 1, 2\} \times \{1, \ldots, 10\}$

In this section, we have seen that when trees are not grown overly large, plugging in $\hat{\sigma}_{\text{SSE}}$ leads to approximate selective Type 1 error control, approximately correct selective coverage, and good power. Unfortunately, providing theoretical guarantees for our procedures when using $\hat{\sigma}_{\text{SSE}}$ would be quite difficult, as the estimator is anti-conservative and depends on the output of CART. Providing theoretical guarantees for our procedures under $\hat{\sigma}_{\text{cons}}$ is more straightforward, using ideas from Gao et al. (2020), Chen and Witten (2022), and Tibshirani et al. (2018). However, as shown in Figure 8, selective Z-tests based on $\hat{\sigma}_{\text{cons}}$ can have very low power. One promising avenue of future work involves providing theoretical guarantees in the regression tree setting for estimators that are less conservative than $\hat{\sigma}_{\text{cons}}$.

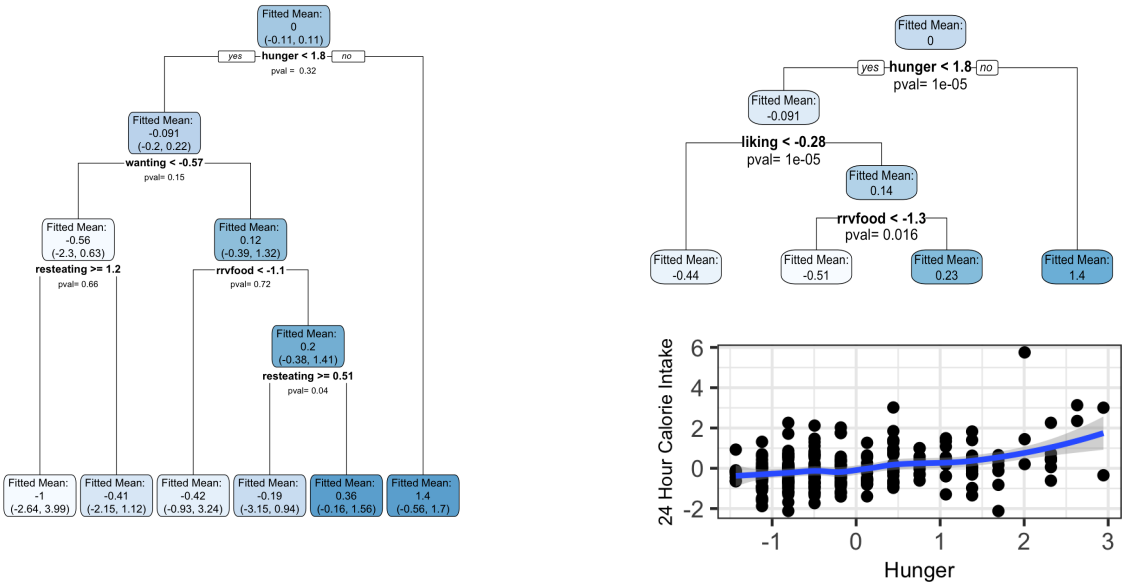

Figure 9: *Left:* A CART tree fit to the Box Lunch Study data. Each split has been labeled with a p-value (8), and each region has been labeled with a confidence interval (23). The shading of the nodes indicates the average response values (white indicates a very small value and dark blue a very large value). *Top right:* A CTree fit to the Box Lunch Study data. *Bottom right:* A scatterplot showing the relationship between the covariate hunger and the response.

6. An Application to the Box Lunch Study

Venkatasubramaniam et al. (2017) compare CART and CTree (Hothorn et al., 2006) within the context of epidemiological studies. They conclude that CTree is preferable to CART because it provides p-values for each split, even though CART has higher predictive accuracy. Since our framework provides p-values for each split in a CART tree, we revisit their analysis of the Box Lunch Study, a clinical trial studying the impact of portion control interventions on 24-hour caloric intake. We consider identifying subgroups of study participants with baseline differences in 24-hour caloric intake on the basis of scores from an assessment that quantifies constructs such as hunger, liking, the relative reinforcement of food (`rrvfood`), and restraint (`resteating`).

We exactly reproduce the trees presented in Figures 1 and 2 of Venkatasubramaniam et al. (2017) by building a CTree using `partykit` and a CART tree using `rpart` on the Box Lunch Study data provided in the `R` package `visTree` (Venkatasubramaniam and Wolfson, 2018). We apply our selective inference framework to compute p-values (8) for each split in CART, and confidence intervals (23) for each region. In this section, we use $\hat{\sigma}_{\text{SSE}}$, defined in Section 5.7, to estimate the error variance. The results are shown in Figure 9.

Both CART and CTree choose `hunger`$<$`1.8` as the first split. For this split, our selective Z-test reports a large p-value of 0.44, while CTree reports a p-value less than 0.001. The conflicting p-values are explained by the difference in null hypotheses. CTree finds strong evidence against the null of no linear association between `hunger` and caloric intake. By contrast, our selective framework for CART does not find strong evidence for a difference between mean caloric intake of participants with `hunger`$<$`1.8` and those with `hunger`$\geq$`1.8`. We see from the bottom right of Figure 9 that while there is evidence of a linear relationship between `hunger` and caloric intake, there is less evidence of a difference in means across the particular split `hunger`=`1.8`. Given that the goal of Venkatasubramaniam et al. (2017) is to "identify population subgroups that are relatively homogeneous with respect to an outcome", the p-value resulting from our selective framework is more natural than the p-value output by CTree, since the former relates directly to the subgroups formed by the split, whereas the latter does not take into account the location of the split point. In general, the left-hand panel of Figure 9 shows that the subgroups of patients identified by CART are not significantly different from one another. This is an important finding that would be missed without our selective inference framework. Furthermore, unlike CTree, our framework provides confidence intervals for the mean response in each subgroup.

An alternative analysis using $\hat{\sigma}_{\text{cons}}$, defined in Section 5.7, is provided in Appendix H, and leads to similar findings.

7. Discussion

Our framework relies on the assumption that $Y \sim N_n(\mu, \sigma^2 I)$, with σ^2 known. In Section 5.7, we showed strong empirical performance when the variance is unknown and σ^2 is estimated. In this section, we briefly comment on the assumptions of spherical variance and normally distributed data.

It natural to wonder whether the assumption that $Y \sim N_n(\mu, \sigma^2 I)$ can be relaxed to the assumption that $Y \sim N_n(\mu, \Sigma)$, with Σ known. Following the work of Lee et al. (2016), the results in Section 3 extend to the setting where $Y \sim N_n(\mu, \Sigma)$ if we:

1. Modify (8) and (15) to condition on the event $\left\{\left(I_n - \frac{\Sigma\nu\nu^T}{\nu^T\Sigma\nu}\right)Y = \left(I_n - \frac{\Sigma\nu\nu^T}{\nu^T\Sigma\nu}\right)y\right\}$ rather than the event $\{\mathcal{P}_\nu^\perp Y = \mathcal{P}_\nu^\perp y\}$, where $\nu = \nu_{sib}$ in the case of (8) and $\nu = \nu_{reg}$ in the case of (15).

2. Replace all instances of the perturbation $y'(\phi, \nu)$, defined in Theorem 6, with the perturbation $y''(\phi, \nu) = \left(I_n - \frac{\Sigma\nu\nu^T}{\nu^T\Sigma\nu}\right)y + \frac{\Sigma\nu}{\nu^T\Sigma\nu}\phi$.

Unfortunately, the modified perturbation $y''(\phi, \nu)$ does not satisfy Condition 1 in Section 4.2 when $\Sigma \neq \sigma^2 I_n$, and so many of the results of Section 4 do not extend to this non-spherical setting. Future work could explore how to efficiently compute the conditioning set in this non-spherical setting.

Furthermore, our framework assumes a normally-distributed response variable. CART is commonly used for classification, survival (Segal, 1988), and treatment effect estimation in causal inference (Athey and Imbens, 2016). While the idea of conditioning on a selection event to control the selective Type 1 error rate applies regardless of the distribution of the response, our Theorem 6 and Theorem 8, and the resulting computational results, relied on

normality of Y. In the absence of this assumption, exactly characterizing the conditioning set and the distribution of the test statistic requires further investigation.

We show in Appendix G that our selective Z-tests approximately control the selective Type 1 error when the normality assumption is violated. Tian and Taylor (2017) and Tibshirani et al. (2018) establish conditions under which selective p-values for linear regression (derived under the assumption of normality) will be asymptotically uniformly distributed under non-normality. Thus suggests the possibility of developing asymptotic theory for our proposed selective Z-tests under violations of normality.

A reviewer pointed out similarities between the problem of testing significance of the first split in the tree and significance testing for a single changepoint, as in Bhattacharya (1994). Building on this connection may provide an avenue for future work.

A software implementation of the methods in this paper is available in the `R` package `treevalues`, at `https://github.com/anna-neufeld/treevalues`.

Acknowledgments

Daniela Witten and Anna Neufeld were supported by the National Institutes of Health (NIH R01 GM12399) and the Simons Foundation (Simons Investigator Award in Mathematical Modeling of Living Systems). Lucy Gao was supported by the Natural Sciences and Engineering Research Council of Canada (Discovery Grants).

Appendix A. Comparison to Loh et al. (2019)

Loh et al. (2016) and Loh et al. (2019) use regression trees to find subgroups of patients with similar treatment effects in clinical trials. They grow trees based on patient characteristics using a different algorithm than CART. Furthermore, they are interested in the mean treatment effect (which is a linear regression coefficient) within each terminal region of the tree, rather than the mean response within each region.

One of the goals of Loh et al. (2019) is to construct valid post-selection confidence intervals for the treatment effect within each terminal node. In this appendix, we show that their approach, when adapted to the setting of this paper, does not yield confidence intervals with nominal coverage.

The basic idea of Loh et al. (2019), instantiated to our setting, is as follows. Suppose that $R_A \in \text{TREE}^{\lambda}(y)$, and define the vector ν_{reg} such that $(\nu_{reg})_i = 1_{(x_i \in R_A)} / \left\{\sum_{i'=1}^{n} 1_{(x_{i'} \in R_A)}\right\}$, as in (14). We know that the "naive" Z-interval does not achieve nominal coverage, meaning that

$$\Pr\left(\nu_{reg}^{\top}\mu \in \left[\nu_{reg}^{\top}y - z_{\alpha/2}\frac{\sigma}{\sqrt{\sum_{i=1}^{n} 1_{(x_i \in R_A)}}}, \nu_{reg}^{\top}y + z_{\alpha/2}\frac{\sigma}{\sqrt{\sum_{i=1}^{n} 1_{(x_i \in R_A)}}}\right]\right) < 1-\alpha. \quad (24)$$

This is because the "multiplier" for the naive confidence interval, $z_{\alpha/2}$, is derived under the assumption that the region R_A (or, equivalently, the vector ν_{reg}), is fixed, rather than a function of the data.

Loh et al. (2019) observe that there exists some $\alpha' < \alpha$ such that

$$\Pr\left(\nu_{reg}^\top \mu \in \left[\nu_{reg}^\top y - z_{\alpha'/2}\frac{\sigma}{\sum_{i=1}^n 1_{(x_i \in R_A)}}, \nu_{reg}^\top y + z_{\alpha'/2}\frac{\sigma}{\sum_{i=1}^n 1_{(x_i \in R_A)}}\right]\right) = 1 - \alpha. \quad (25)$$

The value for α' for a given tree will depend on the number of split covariates p, the number of data points n, the depth of the tree, and the value of λ used for tree pruning, among other considerations. If we know how the data was generated, then we can check whether some value α' satisfies (25) as follows:

1. Draw B different simulated datasets $\{(X_b, y_b)\}_{b=1}^B$ from the same distribution (call this F) as the original data. For $b = 1, \ldots, B$:
   (a) Build a tree using the simulated data (X_b, y_b), using the same procedure and the same settings as in Step 1, and denote it $\text{TREE}^\lambda(y_b)$.
   (b) For each terminal region $R \in \text{TERM}\left(\text{TREE}^\lambda(y_b), \mathbb{R}^p\right)$ in the tree:
      i. Construct a $(1 - \alpha')$ naive Z-interval for the mean response in the region using (X_b, y_b).
      ii. Check if each interval contains $\bar{\mu}_R$, the true mean for this region R.
2. Compute the fraction of intervals in 1(b) that contain $\bar{\mu}_R$.
3. If this value is $1 - \alpha$, then we have found the correct value of α'. If not, then we try a larger or smaller value of α'.

We test this procedure in a very simple simulation study. We generate data $X_{ij} \sim N(0, 1)$ and $y_i \sim N(0, 1)$ for $i = 1, \ldots, 100$ and $j = 1, \ldots, p$. In this simple setting, the true mean response for every region in every fitted tree is 0. We carry out the procedure outlined above with $\alpha = 0.1$. For two values of p and for CART trees with 1, 2, and 3 levels, we create 1000 datasets and 1000 trees and report the empirical coverage of the intervals obtained using this ideal method, averaged over all nodes in all trees. The results, shown in Table 4, show that this ideal procedure procedure leads to intervals that achieve nominal coverage.

Unfortunately, this ideal procedure is practically infeasible, as it requires the user to know the true distribution of the data F. Thus, in practice, Loh et al. (2019) propose replacing F by $\hat{F}$, the empirical distribution of the original data. This amounts to replacing the simulated datasets in Step 2 with bootstrapped datasets, and checking whether the naive Z-intervals in Step 1(b) contain $\bar{y}_R$ rather than $\bar{\mu}_R$.

We can see why this is problematic in a very simple setting where we fit a tree with depth 1. If all observations have mean 0, then a CART tree fit to a bootstrap sample of the data will nevertheless find regions R_L and R_R such that the sample mean value of y_b within R_L is negative and the sample mean value of y_b within R_R is positive. As y_b and y contain many overlapping observations, it is likely that the sample mean value of y within R_L is also negative and the sample mean value of y within R_R is also positive. In other words, because of the overlap between y and y_b, the within-region sample means of y_b are closer to the within-region sample means of y than they are to the within-region population

means. Thus, when we calibrate α' to cover the mean values of y within various regions, we end up with under-coverage of the true population mean, as shown in Table 4.

We see in Table 4 that our selective inference framework approach enables valid inference in this setting, whereas a bootstrap procedure modeled after Loh et al. (2019) does not.

		Loh (ideal)		Loh (bootstrap)		Selective CIs
p	Tree depth	Coverage	Average α'	Coverage	Average α'	Coverage
2	1	0.902	0.008	0.749	0.037	0.890
	2	0.905	0.004	0.695	0.038	0.904
	3	0.900	0.005	0.660	0.047	0.895
20	1	0.883	0.001	0.601	0.016	0.908
	2	0.900	0.00025	0.549	0.016	0.901
	3	0.904	0.00015	0.543	0.022	0.905

Table 4: Coverage of 90% confidence intervals computed using three methods for the simple setting where $y_i \sim N(0,1)$ and $X_{ij} \sim N(0,1)$ for for $i = 1, \ldots, 100$ and $j = 1, \ldots, p$. Note that the "Loh (ideal)" method can never be used in practice, as it requires knowledge of the true parameter.

Appendix B. Proofs for Section 3

B.1 Proof of Theorem 6

Let $0 \leq \alpha \leq 1$. We start by proving the first statement in Theorem 6:

$$pr_{H_0}\left\{p_{sib}(Y) \leq \alpha \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y)\right\} = \alpha.$$

This is a special case of Proposition 3 from Fithian et al. (2014). It follows from the definition of $p_{sib}(Y)$ in (8) that

$$pr_{H_0}\left\{p_{sib}(Y) \leq \alpha \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y), \mathcal{P}^{\perp}_{\nu_{sib}} Y = \mathcal{P}^{\perp}_{\nu_{sib}} y\right\} = \alpha.$$

Therefore, applying the law of total expectation yields

$$\begin{aligned}
&pr_{H_0}\left\{p_{sib}(Y) \leq \alpha \mid R_A, R_B \text{ siblings in } \textsc{tree}^{\lambda}(Y)\right\} \\
&= E_{H_0}\left[1_{\{p_{sib}(Y)<\alpha\}} \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y)\right] \\
&= E_{H_0}\Bigg(E_{H_0}\left[1_{\{p_{sib}(Y)<\alpha\}} \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y), \mathcal{P}^{\perp}_{\nu_{sib}} Y = \mathcal{P}^{\perp}_{\nu_{sib}} y\right] \\
&\qquad\qquad \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y)\Bigg) \\
&= E_{H_0}\left\{\alpha \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y)\right\} = \alpha.
\end{aligned}$$

The second statement of Theorem 6 follows directly from the following result.

Lemma 20 *If $Y \sim N_n\left(\mu, \sigma^2 I_n\right)$, then the random variable $\nu_{sib}^{\mathrm{T}} Y$ has the following conditional distribution:*

$$\nu_{sib}^{\mathrm{T}} Y \mid \{R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y), \mathcal{P}_{\nu_{sib}}^{\perp} Y = \mathcal{P}_{\nu_{sib}}^{\perp} y\} \sim \mathcal{TN}\left\{\nu_{sib}^{\mathrm{T}}\mu, \sigma^2\|\nu_{sib}\|_2^2; S_{sib}^{\lambda}(\nu_{sib})\right\}, \quad (26)$$

where $S_{sib}^{\lambda}(\nu_{sib})$ is defined in (10) and $\mathcal{TN}(\mu, \sigma, S)$ denotes the $N(\mu, \sigma^2)$ distribution truncated to the set S.

Proof

The following holds for any $\nu \in \mathbb{R}^n$.

$$\begin{aligned}
&pr\left\{\nu^{\mathrm{T}} Y > c \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y), \mathcal{P}_{\nu}^{\perp} Y = \mathcal{P}_{\nu}^{\perp} y\right\} \\
&= pr\left\{\nu^{\mathrm{T}} Y > c \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}\left(\mathcal{P}_{\nu}^{\perp} Y + \frac{\nu\nu^{\mathrm{T}}}{\|\nu\|_2^2} Y\right), \mathcal{P}_{\nu}^{\perp} Y = \mathcal{P}_{\nu}^{\perp} y\right\} \\
&= pr\left\{\nu^{\mathrm{T}} Y > c \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}\left(\mathcal{P}_{\nu}^{\perp} y + \frac{\nu}{\|\nu\|_2^2}\nu^{\mathrm{T}} Y\right), \mathcal{P}_{\nu}^{\perp} Y = \mathcal{P}_{\nu}^{\perp} y\right\} \\
&= pr\left\{\nu^{\mathrm{T}} Y > c \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}\left(\mathcal{P}_{\nu}^{\perp} y + \frac{\nu}{\|\nu\|_2^2}\nu^{\mathrm{T}} Y\right)\right\} \\
&= pr\left\{\phi > c \mid \phi \in S_{sib}^{\lambda}(\nu)\right\},
\end{aligned}$$

where $\phi = \nu^{\mathrm{T}} Y$. In the fourth line, the condition $\mathcal{P}_{\nu}^{\perp} Y = \mathcal{P}_{\nu}^{\perp} y$ can be dropped because when $Y \sim N_n\left(\mu, \sigma^2 I_n\right)$, $\mathcal{P}_{\nu}^{\perp} Y$ is independent of $\nu^{\top} Y$. Finally, since $\phi \sim N\left(\nu^{\mathrm{T}}\mu, \sigma^2\|\nu\|_2^2\right)$, (26) holds. ■

B.2 Proof of Proposition 7

Theorem 6.1 from Lee et al. (2016) says that the truncated normal distribution has monotone likelihood ratio in the mean parameter. This guarantees that $L(y)$ and $U(y)$ in (13) are unique. Then, for $L(\cdot)$ and $U(\cdot)$ in (13), (26) in Lemma 20 guarantees that

$$pr\left\{\nu^{\mathrm{T}}\mu \in [L(Y), U(Y)] \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y), \mathcal{P}_{\nu}^{\perp} Y = \mathcal{P}_{\nu}^{\perp} y\right\} = 1 - \alpha. \quad (27)$$

Finally, we need to prove that (27) implies $(1-\alpha)$–*selective coverage* as defined in (12). Following Proposition 3 from Fithian et al. (2014), let η be the random variable $\mathcal{P}_{\nu}^{\perp} Y$ and let $f(\cdot)$ be its density. Then,

$$\begin{aligned}
&pr\left\{\nu^{\mathrm{T}}\mu \in [L(Y), U(Y)] \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y)\right\} \\
&= \int pr\left\{\nu^{\mathrm{T}}\mu \in [L(Y), U(Y)] \mid R_A, R_B \text{ are siblings in } \textsc{tree}^{\lambda}(Y), \mathcal{P}_{\nu}^{\perp} Y = \mathcal{P}_{\nu}^{\perp} y\right\} f(\eta) d\eta \\
&= \int (1-\alpha) f(\eta) d\eta = 1 - \alpha.
\end{aligned}$$

B.3 Proof of Theorem 8

We omit the proof of the first statement of Theorem 8, as it is similar to the proof of the first statement of Theorem 6 in Appendix B.1.

The second statement in Theorem 8 follows directly from the following result.

Lemma 21 *The random variable $\nu_{reg}^{\mathrm{T}} Y$ has the conditional distribution*

$$\nu_{reg}^{\mathrm{T}} Y \mid \{R_A \in \mathrm{TREE}^{\lambda}(Y), \mathcal{P}^{\perp}_{\nu_{reg}} Y = \mathcal{P}^{\perp}_{\nu_{reg}} y\} \sim \mathcal{TN}\left\{\nu_{reg}^{\mathrm{T}}\mu, \sigma^2 \|\nu_{reg}\|_2^2; S^{\lambda}_{reg}(\nu_{reg})\right\}, \quad (28)$$

where $S^{\lambda}_{reg}(\nu_{reg})$ was defined in (16)*.*

We omit the proof of Lemma 21, as it is similar to the proof of Lemma 20.

B.4 Proof of Proposition 9

The proof largely follows the proof of Proposition 7. The fact that the truncated normal distribution has monotone likelihood ratio (Theorem 6.1 of Lee et al. 2016) ensures that $L(y)$ and $U(y)$ defined in (17) are unique, and (28) in Lemma 21 implies that

$$pr\left\{\nu_{reg}^{\mathrm{T}}\mu \in [L(Y), U(Y)] \mid R_A \in \mathrm{TREE}^{\lambda}(Y), \mathcal{P}^{\perp}_{\nu_{reg}} Y = \mathcal{P}^{\perp}_{\nu_{reg}} y\right\} = 1 - \alpha.$$

The rest of the argument is as in the proof of Proposition 7.

Appendix C. Proofs for Section 4.1

C.1 Proof of Lemma 11

We first state and prove the following lemma.

Lemma 22 *Let R_A and R_B be the regions in the definition of ν_{sib} in* (6)*. For an arbitrary region R and for any $j \in \{1, \ldots, p\}$ and $s \in \{1, \ldots, n-1\}$, recall that the potential children of R (the ones that CART will consider adding to the tree when applying Algorithm* A*1 to region R) are given by $R \cap \chi_{j,s,0}$ and $R \cap \chi_{j,s,1}$, where $\chi_{j,s,0}$ and $\chi_{j,s,l}$ were defined in* (1)*. If $(R_A \cup R_B) \subseteq R \cap \chi_{j,s,0}$ or $(R_A \cup R_B) \subseteq R \cap \chi_{j,s,1}$, then $Gain_R\{y'(\phi, \nu_{sib}), j, s\} = Gain_R\{y, j, s\}$ for all ϕ.*

Proof It follows from algebra that for $Gain_R(y, j, s)$ defined in (2),

$$Gain_R(y,j,s) = -\left\{\sum_{i=1}^n 1_{(x_i \in R)}\right\}(\overline{y}_R)^2 + \left\{\sum_{i=1}^n 1_{(x_i \in R \cap \chi_{j,s,0})}\right\}\left(\overline{y}_{R\cap\chi_{j,s,0}}\right)^2 + \left\{\sum_{i=1}^n 1_{(x_i \in R \cap \chi_{j,s,1})}\right\}\left(\overline{y}_{R\cap\chi_{j,s,1}}\right)^2, \quad (29)$$

where $\overline{y}_R = \left(\sum_{i\in R} y_i\right) / \left\{\sum_{i=1}^n 1_{(x_i \in R)}\right\}$. It follows from (29) that to prove Lemma 22, it suffices to show that $\overline{y}_T = \overline{y'(\phi, \nu_{sib})}_T$ for $T \in \{R, R \cap \chi_{j,s,0}, R \cap \chi_{j,s,1}\}$. Recall from Section 3.3 that $\{y'(\phi, \nu_{sib})\}_i = y_i + \Delta_i$, where

$$\Delta_i = \begin{cases} (\phi - \nu_{sib}^{\mathrm{T}} y)\frac{\sum_{i'=1}^n 1_{(x'_i \in R_B)}}{\sum_{i'=1}^n 1_{(x'_i \in R_A \cup R_B)}} & \text{if } i \in R_A \\ -(\phi - \nu_{sib}^{\mathrm{T}} y)\frac{\sum_{i'=1}^n 1_{(x'_i \in R_A)}}{\sum_{i'=1}^n 1_{(x'_i \in R_A \cup R_B)}} & \text{if } i \in R_B \\ 0 & \text{otherwise.} \end{cases}$$

Without loss of generality, assume that $(R_A \cup R_B) \subseteq R \cap \chi_{j,s,0}$. For any $T \in \{R, R \cap \chi_{j,s,0}\}$, $R_A \cup R_B \subseteq T$. Thus,

$$\begin{aligned}
\overline{y'(\phi, \nu_{sib})}_T &= \frac{1}{\sum_{i=1}^n 1_{(x_i \in T)}} \left\{ \sum_{i \in T \setminus (R_A \cup R_B)} y_i + \sum_{i \in R_A} (y_i + \Delta_i) + \sum_{i \in R_B} (y_i + \Delta_i) \right\} \\
&= \bar{y}_T + \frac{\sum_{i \in R_A} \Delta_i + \sum_{i \in R_B} \Delta_i}{\sum_{i=1}^n 1_{(x_i \in T)}} \\
&= \bar{y}_T + \frac{\left\{\sum_{i=1}^n 1_{(x_i \in R_A)}\right\} (\phi - \nu_{sib}^{\mathrm{T}} y) \frac{\sum_{i=1}^n 1_{(x_i \in R_B)}}{\sum_{i=1}^n 1_{(x_i \in R_A \cup R_B)}} - \left\{\sum_{i=1}^n 1_{(x_i \in R_B)}\right\} (\phi - \nu_{sib}^{\mathrm{T}} y) \frac{\sum_{i=1}^n 1_{(x_i \in R_A)}}{\sum_{i=1}^n 1_{(x_i \in R_A \cup R_B)}}}{\sum_{i=1}^n 1_{(x_i \in T)}} \\
&= \bar{y}_T + 0 = \bar{y}_T.
\end{aligned}$$

Furthermore,

$$\overline{y'(\phi, \nu_{sib})}_{R \cap \chi_{j,s,1}} = \frac{1}{\sum_{i=1}^n 1_{(x_i \in R \cap \chi_{j,s,1})}} \sum_{i \in R \cap \chi_{j,s,1}} (y_i + \Delta_i) = \frac{1}{\sum_{i=1}^n 1_{(x_i \in R \cap \chi_{j,s,1})}} \sum_{i \in R \cap \chi_{j,s,1}} (y_i + 0) = \overline{y}_{R \cap \chi_{j,s,1}}.$$

■

We will now prove Lemma 11.

It follows from Definition 10 that if $\mathcal{R}\left[\textsc{branch}\{R, \textsc{tree}^\lambda(y)\}\right] \subseteq \textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$, then R_A and R_B are siblings in $\textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$. This establishes the ($\Leftarrow$) direction.

We will prove the ($\Rightarrow$) direction by contradiction. Suppose that R_A and R_B are siblings in $\textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$. Define $\textsc{branch}\{R_A, \textsc{tree}^\lambda(y)\} = ((j_1, s_1, e_1), \ldots, (j_L, s_L, e_L))$, and define $R^{(l')} = \bigcap_{l=1}^{l'} \chi_{j_l, s_l, e_l}$ for $l' = 1, \ldots, L$. Assume that there exists $l \in \{0, \ldots, L-2\}$ such that $R^{(l)} \in \textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$ and $R^{(l+1)} \notin \textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$. We assume that any ties between splits that occur at Step 2 of Algorithm A1 are broken in the same way for y and $y'(\phi, \nu_{sib})$, and so this implies that there exists $(\tilde{j}, \tilde{s}) \neq (j_{l+1}, s_{l+1})$ such that $(\tilde{j}, \tilde{s}) \in \arg\max_{j,s} Gain_{R^{(l)}}\{y'(\phi, \nu_{sib}), j, s\}$ and

$$Gain_{R^{(l)}}\{y'(\phi, \nu_{sib}), j_{l+1}, s_{l+1}\} < Gain_{R^{(l)}}\{y'(\phi, \nu_{sib}), \tilde{j}, \tilde{s}\}. \tag{30}$$

Since R_A and R_B are siblings in $\textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$, it follows from Lemma 22 that $Gain_{R^{(l)}}\{y'(\phi, \nu_{sib}), \tilde{j}, \tilde{s}\} = Gain_{R^{(l)}}(y, \tilde{j}, \tilde{s})$. Also, since R_A and R_B are siblings in $\textsc{tree}^\lambda(y)$, it follows from Lemma 22 that $Gain_{R^{(l)}}\{y'(\phi, \nu_{sib}), j_{l+1}, s_{l+1}\} = Gain_{R^{(l)}}(y, j_{l+1}, s_{l+1})$. Applying these facts to (30) yields

$$Gain_{R^{(l)}}(y, j_{l+1}, s_{l+1}) < Gain_{R^{(l)}}(y, \tilde{j}, \tilde{s}). \tag{31}$$

But since $R^{(l)}$ and $R^{(l+1)}$ both appeared in $\textsc{tree}^\lambda(y)$,

$$(j_{l+1}, s_{l+1}) \in \arg\max_{j,s} Gain_{R^{(l)}}(y, j, s).$$

This contradicts (31). Therefore, for any $l \in \{0, \ldots, L-2\}$, if $R^{(l)} \in \textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$, then $R^{(l+1)} \in \textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$. Since $R^{(0)} \in \textsc{tree}^\lambda\{y'(\phi, \nu_{sib})\}$, the proof follows by induction.

C.2 Proof of Lemma 13

Let $R_A \in \text{TREE}^\lambda(y)$ with $\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\} = ((j_1, s_1, e_1), \ldots, (j_L, s_L, e_L))$ such that $R_A = \bigcap_{l=1}^{L} \chi_{j_l, s_l, e_l}$. Since Algorithm A1 creates regions by intersecting halfspaces and set intersections are invariant to the order of intersection, it follows that $R_A = \bigcap_{l=1}^{L} \chi_{j_l, s_l, e_l} \in \text{TREE}^\lambda\{y'(\phi, \nu_{reg})\}$ if and only if there exists $\pi \in \Pi$ such that

$$\left\{ \bigcap_{l=1}^{l'} \chi_{j_{\pi(l)}, s_{\pi(l)}, e_{\pi(l)}} \right\}_{l'=1}^{L} \subseteq \text{TREE}^\lambda\{y'(\phi, \nu_{reg})\}.$$

By Definitions 10 and 12,

$$\mathcal{R}(\pi[\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}]) = \left\{ \bigcap_{l=1}^{l'} \chi_{j_{\pi(l)}, s_{\pi(l)}, e_{\pi(l)}} \right\}_{l'=1}^{L}.$$

Thus,

$$\begin{aligned} S^\lambda_{reg} &= \left\{ \phi : R_A \in \text{TREE}^\lambda\{y'(\phi, \nu_{reg})\} \right\} \\ &= \bigcup_{\pi \in \Pi} \left\{ \phi : \mathcal{R}(\pi[\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}]) \subseteq \text{TREE}^\lambda \left\{ y'(\phi, \nu_{reg}) \right\} \right\} \\ &= \bigcup_{\pi \in \Pi} S^\lambda \left(\pi \left[\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\} \right], \nu_{reg} \right), \end{aligned}$$

where the third equality follows from the definition of $S^\lambda(\mathcal{B}, \nu)$ in (18).

Appendix D. Proofs for Section 4.2

D.1 Proof of Proposition 14

Recall that $\mathcal{B} = ((j_1, s_1, e_1), \ldots, (j_L, s_L, e_L))$ and $\mathcal{R}(\mathcal{B}) = \{R^{(0)}, \ldots, R^{(L)}\}$. Recall from (20) that $S_{grow}(\mathcal{B}, \nu) = \{\phi : \mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^0\{y'(\phi, \nu)\}\}$, and that we define $S_{l,j,s} = \left\{\phi : \text{GAIN}_{R^{(l-1)}}\{y'(\phi, \nu), j, s\} \leq \text{GAIN}_{R^{(l-1)}}\{y'(\phi, \nu), j_l, s_l\}\right\}$.

For $l = 1, \ldots, L$,

$$R^{(l-1)} \in \text{TREE}^0\{y'(\phi, \nu)\} \text{ and } \phi \in \cap_{s=1}^{n-1} \cap_{j=1}^{p} S_{l,j,s} \iff \{R^{(l-1)}, R^{(l)}\} \subseteq \text{TREE}^0\{y'(\phi, \nu)\}, \quad (32)$$

because, given that $R^{(l-1)} \in \text{TREE}^0\{y'(\phi, \nu)\}$, $R^{(l)} \in \text{TREE}^0\{y'(\phi, \nu)\}$ if and only if $(j_l, s_l) \in \arg\max_{(j,s): s \in \{1,\ldots,n-1\}, j \in \{1,\ldots,p\}} \text{GAIN}_{R^{(l-1)}}\left(y'(\phi, \nu), j, s\right)$. Combining (32) with the fact that $\{\phi : R^{(0)} \in \text{TREE}^0\{y'(\phi, \nu)\}\} = \mathbb{R}$ yields

$$\bigcap_{l=1}^{L} \bigcap_{j=1}^{p} \bigcap_{s=1}^{n-1} S_{l,j,s} = \bigcap_{l=1}^{L} \left\{ \phi : \{R^{(l-1)}, R^{(l)}\} \subseteq \text{TREE}^0\{y'(\phi, \nu)\} \right\} = \{\phi : \mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^0\{y'(\phi, \nu)\}\}.$$

D.2 Proof of Proposition 15

Given a region R, let $\mathbb{1}(R)$ denote the vector in $\mathbb{R}^n$ such that the ith element is $1_{(x_i \in R)}$. Let $\mathcal{P}_{\mathbb{1}(R)} = \mathbb{1}(R)\left\{\mathbb{1}(R)^{\mathrm{T}}\mathbb{1}(R)\right\}^{-1}\mathbb{1}(R)^{\mathrm{T}}$ denote the orthogonal projection matrix onto the vector $\mathbb{1}(R)$.

Lemma 23 *For any region R, $\text{GAIN}_R(y,j,s) = y^{\mathrm{T}} M_{R,j,s} y$, where*

$$M_{R,j,s} = \mathcal{P}_{\mathbb{1}(R\cap\chi_{j,s,1})} + \mathcal{P}_{\mathbb{1}(R\cap\chi_{j,s,0})} - \mathcal{P}_{\mathbb{1}(R)}. \tag{33}$$

Furthermore, the matrix $M_{R,j,s}$ is positive semidefinite.

Proof For any region R, $\sum_{i\in R}(y_i - \bar{y}_R)^2 = \sum_{i\in R} y_i^2 - y^{\mathrm{T}}\mathcal{P}_{\mathbb{1}_{(R)}} y$. Thus, from (2),

$$\begin{aligned}
\text{GAIN}_R(y,j,s) &= \sum_{i\in R}(y_i - \bar{y}_R)^2 - \sum_{i\in R\cap\chi_{j,s,1}}(y_i - \bar{y}_{R\cap\chi_{j,s,1}})^2 - \sum_{i\in R\cap\chi_{j,s,0}}(y_i - \bar{y}_{R\cap\chi_{j,s,0}})^2 \\
&= \sum_{i\in R} y_i^2 - y^{\mathrm{T}}\mathcal{P}_{\mathbb{1}_{(R)}} y - \sum_{i\in R\cap\chi_{j,s,1}} y_i^2 + y^{\mathrm{T}}\mathcal{P}_{\mathbb{1}(R\cap\chi_{j,s,1})} y - \sum_{i\in R\cap\chi_{j,s,0}} y_i^2 + y^{\mathrm{T}}\mathcal{P}_{\mathbb{1}(R\cap\chi_{j,s,0})} y \\
&= y^{\mathrm{T}}\left\{\mathcal{P}_{\mathbb{1}(R\cap\chi_{j,s,1})} + \mathcal{P}_{\mathbb{1}(R\cap\chi_{j,s,0})} - \mathcal{P}_{\mathbb{1}(R)}\right\} y = y^{\mathrm{T}} M_{R,j,s} y.
\end{aligned}$$

To see that $M_{R,j,s}$ is positive semidefinite, observe that, for any vector v,

$$\begin{aligned}
v^{\mathrm{T}} M_{R,j,s} v = \text{GAIN}_R(v,j,s) &= \sum_{i\in R}(v_i - \bar{v}_R)^2 - \min_{a_1,a_2}\left\{\sum_{i\in R\cap\chi_{j,s,1}}(v_i - a_1)^2 + \sum_{i\in R\cap\chi_{j,s,0}}(v_i - a_2)^2\right\} \\
&\geq \sum_{i\in R}(v_i - \bar{v}_R)^2 - \left\{\sum_{i\in R\cap\chi_{j,s,1}}(v_i - \bar{v}_R)^2 + \sum_{i\in R\cap\chi_{j,s,0}}(v_i - \bar{v}_R)^2\right\} = 0.
\end{aligned}$$

■

It follows from Lemma 23 that we can express each set $S_{l,j,s}$ from Proposition 14 as

$$\begin{aligned}
S_{l,j,s} &= \left\{\phi : \text{GAIN}_{R^{(l-1)}}\{y'(\phi,\nu), j, s\} \leq \text{GAIN}_{R^{(l-1)}}\{y'(\phi,\nu), j_l, s_l\}\right\} \\
&= \left\{\phi : y'(\phi,\nu)^{\mathrm{T}} M_{R^{(l-1)},j,s} y'(\phi,\nu) \leq y'(\phi,\nu)^{\mathrm{T}} M_{R^{(l-1)},j_l,s_l} y'(\phi,\nu)\right\}.
\end{aligned} \tag{34}$$

We now use (34) to prove the first statement of Proposition 15.

Lemma 24 *Each set $S_{l,j,s}$ is defined by a quadratic inequality in ϕ.*

Proof The definition of $y'(\phi,\nu)$ in (10) implies that

$$\begin{aligned}
y'(\phi,\nu)^{\mathrm{T}} M_{R,j,s} y'(\phi,\nu) &= \left(\mathcal{P}_\nu^\perp y + \frac{\nu\phi}{\|\nu\|_2^2}\right)^{\mathrm{T}} M_{R,j,s}\left(\mathcal{P}_\nu^\perp y + \frac{\nu\phi}{\|\nu\|_2^2}\right) \\
&= \frac{\nu^{\mathrm{T}} M_{R,j,s}\nu}{\|\nu\|_2^4}\phi^2 + \frac{2\nu^{\mathrm{T}} M_{R,j,s}\mathcal{P}_\nu^\perp y}{\|\nu\|_2^2}\phi + y^{\mathrm{T}} P_\nu^\perp M_{R,j,s} P_\nu^\perp y \\
&\equiv a(R,j,s)\phi^2 + b(R,j,s)\phi + c(R,j,s).
\end{aligned} \tag{35}$$

Therefore, by (34),

$$\begin{aligned}
S_{l,j,s} = \Bigg\{\phi : &\left[a\left\{R^{(l-1)}, j, s\right\} - a\left\{R^{(l-1)}, j_l, s_l\right\}\right]\phi^2 + \left[b\left\{R^{(l-1)}, j, s\right\} - b\left\{R^{(l-1)}, j_l, s_l\right\}\right]\phi \\
&+ \left[c\left\{R^{(l-1)}, j, s\right\} - c\left\{R^{(l-1)}, j_l, s_l\right\}\right] \leq 0\Bigg\}.
\end{aligned} \tag{36}$$

■

Proposition 14 indicates that to compute $S_{grow}(\mathcal{B}, \nu)$ from (20), we need to compute the coefficients of the quadratic for each $S_{l,j,s}$, where $l = 1, \ldots, L$, $j = 1, \ldots, p$, and $s = 1, \ldots, n-1$.

Lemma 25 *We can compute the coefficients* $a\left\{R^{(l-1)}, j, s\right\}, b\left\{R^{(l-1)}, j, s\right\}$ *and* $c\left\{R^{(l-1)}, j, s\right\}$, *defined in Lemma 24, for* $l = 1, \ldots, L$, $j = 1, \ldots, p$, *and* $s = 1, \ldots, n-1$, *in* $O\left\{nplog(n) + npL\right\}$ *operations.*

Proof

Using the definitions in Lemmas 23 and 24 and algebra, we have that

$$\|\nu\|_2^4 a\{R^{(l-1)}, j, s\} = \frac{\left[\nu^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,1}\}\right]^2}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,1}\}}} + \frac{\left[\nu^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,0}\}\right]^2}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,0}\}}} - \frac{\left[\nu^{\mathrm{T}} \mathbb{1}\{R^{(l-1)}\}\right]^2}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)}\}}}, \tag{37}$$

$$\begin{aligned}\frac{1}{2}\|\nu\|_2^2 b\{R^{(l-1)}, j, s\} = &\frac{\nu^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,1}\} \left(\mathcal{P}_\nu^\perp y\right)^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,1}\}}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,1}\}}} + \\ &\frac{\nu^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,0}\} \left(\mathcal{P}_\nu^\perp y\right)^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,0}\}}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,0}\}}} - \frac{\nu^{\mathrm{T}} \mathbb{1}\{R^{(l-1)}\} \left(\mathcal{P}_\nu^\perp y\right)^{\mathrm{T}} \mathbb{1}\{R^{(l-1)}\}}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)}\}}},\end{aligned} \tag{38}$$

$$c\{R^{(l-1)}, j, s\} = \frac{\left[(\mathcal{P}_\nu^\perp y)^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,1}\}\right]^2}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,1}\}}} + \frac{\left[(\mathcal{P}_\nu^\perp y)^{\mathrm{T}} \mathbb{1}\{R^{(l-1)} \cap \chi_{j,s,0}\}\right]^2}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,0}\}}} - \frac{\left[(\mathcal{P}_\nu^\perp y)^{\mathrm{T}} \mathbb{1}\{R^{(l-1)}\}\right]^2}{\sum_{i=1}^n 1_{\{i \in R^{(l-1)}\}}}. \tag{39}$$

We compute the scalar $\|\nu\|_2^2$ and the vector $\mathcal{P}_\nu^\perp y$ in $O(n)$ operations once at the start of the algorithm. We also sort each feature in $O\left[nlog(n)\right]$ operations per feature. We will now show that for the lth level and the jth feature, we can compute $a\{R^{(l-1)}, j, s\}$, $b\{R^{(l-1)}, j, s\}$, and $c\{R^{(l-1)}, j, s\}$ for all $n-1$ values of s in $O(n)$ operations.

The index s appears in (37)–(39) only through $\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,1}\}}$, $\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,0}\}}$, and through inner products of vectors ν and $\mathcal{P}_\nu^\perp y$ with indicator vectors $\mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,s,1}\right\}$ and $\mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,s,0}\right\}$. For simplicity, we assume that covariate x_j is continuous, and thus the order statistics are unique.

Let m_1 be the index corresponding to the smallest value of x_j. Then $\nu^{\mathrm{T}} \mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,1,1}\right\} = \nu_{m_1}$ if observation m_1 is in $R^{(l-1)}$, and is 0 otherwise. Similarly, $\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,1,1}\}} = 1$ if observation m_1 is in $R^{(l-1)}$, and is 0 otherwise. Next, let m_2 be the index corresponding to the second smallest value of x_j. Then $\nu^{\mathrm{T}} \mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,2,1}\right\} = \mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,1,1}\right\} + \nu_{m_2}$ if observation m_2 is in $R^{(l-1)}$, and is equal to $\mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,1,1}\right\}$ otherwise. We compute $\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,2,1}\}}$ in the same manner. Each update is done in constant time. Continuing in this manner, computing the full set of $n-1$ quantities $\nu^{\mathrm{T}} \mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,s,1}\right\}$ and $\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,1}\}}$ for $s = 1, \ldots, n-1$ requires a single forward pass through the sorted values of x_j, which takes $O(n)$ operations. The same ideas can be applied to compute $\left(\mathcal{P}_\nu^\perp y\right)^{\mathrm{T}} \mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,1,1}\right\}$, $\nu^{\mathrm{T}} \mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,s,0}\right\}$, $\left(\mathcal{P}_\nu^\perp y\right)^{\mathrm{T}} \mathbb{1}\left\{R^{(l-1)} \cap \chi_{j,s,0}\right\}$, and $\sum_{i=1}^n 1_{\{i \in R^{(l-1)} \cap \chi_{j,s,0}\}}$ using constant time updates for each value of s.

Thus, we can obtain all components of coefficients $a\{R^{(l-1)}, j, s\}$, $b\{R^{(l-1)}, j, s\}$, and $c\{R^{(l-1)}, j, s\}$ for a fixed j and l, and for all $s = 1, \ldots, n-1$, in $O(n)$ operations. These scalar components can be combined to obtain the coefficients in $O(n)$ operations. Therefore, given the sorted features, we compute the $(n-1)pL$ coefficients in $O(npL)$ operations. ■

Once the coefficients on the right hand side of (36) have been computed, we can compute $S_{l,j,s}$ in constant time via the quadratic equation: it is either a single interval or the union of two intervals. Finally, in general we can intersect $(n-1)pL$ intervals in $O\{npL \times log(npL)\}$ operations (Bourgon, 2009). The final claim of Proposition 15 involves the special case where $\nu = \nu_{sib}$ and $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$.

Lemma 26 *Suppose that $\nu = \nu_{sib}$ from (6) and $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$. (i) If $l < L$, then for all j and s, there exist $a, b \in [-\infty, \infty]$ such that $a \leq \nu^{\text{T}} y \leq b$ and $S_{l,j,s} = (a, b)$. (ii) If $l = L$, then for all j and s, there exist $c, d \in \mathbb{R}$ such that $c \leq 0 \leq d$ and $S_{l,j,s} = (-\infty, c] \cup [d, \infty)$. (iii) We can intersect all $(n-1)pL$ sets of the form $S_{l,j,s}$ in $O(npL)$ operations.*

Proof This proof relies on the form of $S_{l,j,s}$ given in (36). To prove (i), note that when $l < L$,

$$\begin{aligned}\|\nu_{sib}\|_2^4 \left[a\left\{R^{(l-1)}, j, s\right\} - a\left\{R^{(l-1)}, j_l, s_l\right\}\right] &= \nu_{sib}^{\text{T}} M_{R^{(l-1)},j,s} \nu_{sib} - \nu_{sib}^{\text{T}} M_{R^{(l-1)},j_l,s_l} \nu_{sib} \\ &= \nu_{sib}^{\text{T}} M_{R^{(l-1)},j,s} \nu_{sib} \geq 0.\end{aligned}$$

The first equality follows directly from the definition of $a(R, j, s)$ in (35). To see why the second equality holds, observe that $R_A \cup R_B \subseteq R^{(l-1)}$, and without loss of generality assume that $R_A \cup R_B \subseteq R^{(l-1)} \cap \chi_{j_l,s_l,1}$. Recall that the ith element of ν_{sib} is non-zero if and only if $i \in R_A \cup R_B$, and that the non-zero elements of ν_{sib} sum to 0. Thus, $\mathbb{1}\left\{R^{(l-1)}\right\}^{\text{T}} \nu_{sib} = 0$ and $\mathbb{1}\left\{R^{(l-1)} \cap \chi_{j_l,s_l,1}\right\}^{\text{T}} \nu_{sib} = 0$. Furthermore, the supports of $R^{(l-1)} \cap \chi_{j_l,s_l,0}$ and ν_{sib} are non-overlapping, and so $\mathbb{1}\left\{R^{(l-1)} \cap \chi_{j_l,s_l,0}\right\}^{\text{T}} \nu_{sib} = 0$. Thus,

$$M_{R^{(l-1)},j_l,s_l} \nu_{sib} = \left[\mathcal{P}_{\mathbb{1}\left\{R^{(l-1)} \cap \chi_{j_l,s_l,1}\right\}} + \mathcal{P}_{\mathbb{1}\left\{R^{(l-1)} \cap \chi_{j_l,s_l,0}\right\}} - \mathcal{P}_{\mathbb{1}\left\{R^{(l-1)}\right\}}\right] \nu_{sib} = 0.$$

The final inequality follows because $M_{R^{(l-1)},j,s}$ is positive semidefinite (Lemma 23).

Thus, when $l < L$, $S_{l,j,s}$ is defined in (36) by a quadratic inequality with a non-negative quadratic coefficient. Thus, $S_{l,j,s}$ must be a single interval of the form (a, b). Furthermore, since $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$, we know that $\mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^0(y) = \text{TREE}^0\{y'(\nu^{\text{T}} y, \nu)\}$. Therefore, $\nu^{\text{T}} y \in S_{grow}(\mathcal{B}, \nu_{sib}) = \cap_{l=1}^{L} \cap_{j=1}^{p} \cap_{s=1}^{n-1} S_{l,j,s}$, and so we conclude $a \leq \nu^{\text{T}} y \leq b$. This completes the proof of (i).

To prove (ii), we first prove that when $l = L$ the quadratic equation in ϕ defined in (36) has a non-positive quadratic coefficient. To see this, note that

$$\begin{aligned}&\|\nu_{sib}\|_2^4 \left[a\left\{R^{(L-1)}, j, s\right\} - a\left\{R^{(L-1)}, j_L, s_L\right\}\right] = \nu_{sib}^{\text{T}} M_{R^{(L-1)},j,s} \nu_{sib} - \nu_{sib}^{\text{T}} M_{R^{(L-1)},j_L,s_L} \nu_{sib} \\ &= \nu_{sib}^{\text{T}} \left[\mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,1}\}} + \mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,0}\}}\right] \nu_{sib} - \nu_{sib}^{\text{T}} \left[\mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j_L,s_L,1}\}} + \mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j_L,s_L,0}\}}\right] \nu_{sib} \\ &= \nu_{sib}^{\text{T}} \left[\mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,1}\}} + \mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,0}\}}\right] \nu_{sib} - \nu_{sib}^{\text{T}} \nu_{sib} \\ &= \left\| \left[\mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,1}\}} + \mathcal{P}_{\mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,0}\}}\right] \nu_{sib} \right\|_2^2 - \|\nu_{sib}\|_2^2 \leq 0. \qquad (40)\end{aligned}$$

The first equality follows from (35). The second follows from the definition of $M_{R,j,s}$ given in (33) and from the fact that $\mathcal{P}_{\mathbb{1}\{R^{(L-1)}\}}\nu_{sib} = 0$ because $\mathbb{1}\{R^{(L-1)}\}^{\mathrm{T}}\nu_{sib}$ sums up all of the non-zero elements of ν_{sib}, which sum to 0. The third equality follows because ν_{sib} lies in $span\left[\mathbb{1}\left\{R^{(L-1)} \cap \chi_{j_L,s_L,1}\right\}, \mathbb{1}\left\{R^{(L-1)} \cap \chi_{j_L,s_L,0}\right\}\right]$; projecting it onto this span yields itself. Noting that $\left[\mathcal{P}_{\mathbb{1}\{R^{(L-1)}\cap\chi_{j,s,1}\}} + \mathcal{P}_{\mathbb{1}\{R^{(L-1)}\cap\chi_{j,s,0}\}}\right]$ is itself a projection matrix, the fourth equality follows from the idempotence of projection matrices, and the inequality follows from the fact that $\|\nu_{sib}\|_2 \geq \|Q\nu_{sib}\|_2$ for any projection matrix Q. Thus, when $l = L$, the quadratic that defines $S_{l,j,s}$ has a non-positive quadratic coefficient.

Equality is attained in (40) if and only if $\nu_{sib} \in span\left[\mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,1}\}, \mathbb{1}\{R^{(L-1)} \cap \chi_{j,s,0}\}\right]$. This can only happen if splitting $R^{(L-1)}$ on j, s yields an identical partition of the data to splitting on j_L, s_L. If this is the case, then $S_{L,j,s} = (-\infty, 0] \cup [0, \infty)$ from the definition of $S_{l,j,s}$ in Proposition 14, and so (ii) is satisfied with $c = d = 0$.

We now proceed to the setting where the inequality in (40) is strict. In this case, (36) implies that $S_{L,j,s} = (-\infty, c] \cup [d, \infty)$ for $c \leq d$ and $c, d \in \mathbb{R}$. To complete the proof of (ii), we must argue that $c \leq 0$ and $d \geq 0$. Recall that the quadratic in (34) has the form $\textsc{gain}_{R^{(L-1)}}\{y'(\phi,\nu), j, s\} - \textsc{gain}_{R^{(L-1)}}\{y'(\phi,\nu), j_L, s_L\}$. When $\phi = 0$, $\textsc{gain}_{R^{(L-1)}}\{y'(\phi,\nu), j_L, s_L\} = 0$, because $\phi = 0$ eliminates the contrast between R_A and R_B, so that the split on j_L, s_L provides zero gain. So, when $\phi = 0$, the quadratic evaluates to $\textsc{gain}_{R^{(L-1)}}\{y'(\phi,\nu), j, s\}$, which is non-negative by Lemma 23. Thus, $S_{l,j,s}$ is defined by a downward facing quadratic that is non-negative when $\phi = 0$, and so the set $S_{l,j,s}$ has the form $(-\infty, c] \cup [d, \infty)$ for $c \leq 0 \leq d$.

To prove (iii), observe that (i) implies that $\cap_{l=1}^{L-1} \cap_{j=1}^{p} \cap_{s=1}^{n-1} S_{l,j,s} = (a_{max}, b_{min})$, where a_{max} is the maximum over all of the a's, and b_{min} is the minimum over all of the b's. This can be computed in $np(L-1)$ steps. Furthermore, (ii) implies that $\cap_{j=1}^{p} \cap_{s=1}^{n-1} S_{L,j,s} = (-\infty, c_{min}] \cap [d_{max}, \infty)$, where c_{min} and d_{max} are the minimum over all of the c's and the maximum over all the d's, respectively. This can be computed in np steps. Thus, we can compute $\cap_{l=1}^{L} \cap_{j=1}^{p} \cap_{s=1}^{n-1} S_{l,j,s}$ in $O(npL)$ operations. ■

D.3 Proof of Proposition 16

To prove Proposition 16, we first propose a particular method of constructing an example of $\textsc{tree}(\mathcal{B}, \nu, \lambda)$. We then show that (21) holds for this particular choice for $\textsc{tree}(\mathcal{B}, \nu, \lambda)$. We conclude by evaluating the computational cost of computing such an example of $\textsc{tree}(\mathcal{B}, \nu, \lambda)$, and by arguing that in the special case where $\mathcal{R}(\mathcal{B}) \in \textsc{tree}^{\lambda}(y)$, our example is equal to $\textsc{tree}^{\lambda}(y)$.

When Algorithm A2 is called with parameters $\textsc{tree}, y, \lambda, \mathcal{O}$, where $\mathcal{O}$ is a bottom-up ordering of the K nodes in $\textsc{tree}$, it computes a sequence of intermediate trees, $\textsc{tree}_0, \ldots, \textsc{tree}_K$. We use the notation $\textsc{tree}_k(\textsc{tree}, y, \lambda, \mathcal{O})$, for $k = 0, \ldots, K$, to denote the kth of these intermediate trees. The following lemma helps build up to our proposed example of $\textsc{tree}(\mathcal{B}, \nu, \lambda)$.

Lemma 27 *Let $\phi_1 \in S_{grow}(\mathcal{B}, \nu)$ and $\phi_2 \in S_{grow}(\mathcal{B}, \nu)$. Then $\textsc{tree}^0\{y'(\phi_1, \nu)\} = \textsc{tree}^0\{y'(\phi_2, \nu)\}$. Let $\mathcal{O}$ be a bottom-up ordering of the K regions in $\textsc{tree}^0\{y'(\phi_1, \nu)\}$ such that the last L regions in the ordering are $R^{(L-1)}, \ldots, R^{(0)}$. Then $\textsc{tree}_{K-L}[\textsc{tree}^0\{y'(\phi_1, \nu)\}, y'(\phi_1, \nu), \lambda, \mathcal{O}] = \textsc{tree}_{K-L}[\textsc{tree}^0\{y'(\phi_2, \nu)\}, y'(\phi_2, \nu), \lambda, \mathcal{O}]$.*

Proof We first prove that $\text{TREE}^0\{y'(\phi_1,\nu)\} \subseteq \text{TREE}^0\{y'(\phi_2,\nu)\}$, where $\phi_1, \phi_2 \in S_{grow}(\mathcal{B},\nu)$. The fact that $\phi_1 \in S_{grow}(\mathcal{B},\nu)$ and $\phi_2 \in S_{grow}(\mathcal{B},\nu)$ implies two properties:

Property 1: $R^{(l)} \in \text{TREE}^0\{y'(\phi_1,\nu)\}$ and $R^{(l)} \in \text{TREE}^0\{y'(\phi_2,\nu)\}$ for $l \in \{0,\dots,L\}$ by the definition of $S_{grow}(\mathcal{B},\nu)$.

Property 2: $R^{(l)}_{sib} \in \text{TREE}^0\{y'(\phi_1,\nu)\}$ and $R^{(l)}_{sib} \in \text{TREE}^0\{y'(\phi_2,\nu)\}$ for $l \in \{1,\dots,L\}$, where $R^{(l)}_{sib} \equiv R^{(l-1)} \cap \chi_{j_l,s_l,1-e_l}$. This follows from Property 1 and Definition 1.

Suppose that $R \in \text{TREE}^0\{y'(\phi_1,\nu)\}$. Then R must belong to one of these three cases, illustrated in Figure 10(a):

Case 1: $\exists\ l \in \{0,\dots L\}$ *such that* $R = R^{(l)}$. By Property 1, $R \in \text{TREE}^0\{y'(\phi_2,\nu)\}$.

Case 2: $\exists\ l \in \{1,\dots,L\}$ *such that* $R = R^{(l)}_{sib}$. By Property 2, $R \in \text{TREE}^0\{y'(\phi_2,\nu)\}$.

Case 3: $R \in \text{DESC}[R', \text{TREE}^0\{y'(\phi_1,\nu)\}]$, *where either* $R' = R^{(l)}_{sib}$ *for some* $l \in \{1,\dots,L\}$, *or else* $R' = R^{(L)}$. By Properties 1 and 2, $R' \in \text{TREE}^0\{y'(\phi_2,\nu)\}$. Condition 1 ensures that, for all $i \in R'$ and for some constants c and d,

$$\{y'(\phi_2,\nu)\}_i = \begin{cases} \{y'(\phi_1,\nu)\}_i & \text{if } R' = R^{(l)}_{sib} \text{ for some } l \in \{1,\dots,L-1\}, \\ \{y'(\phi_1,\nu)\}_i + c & \text{if } R' = R^{(L)}_{sib}, \\ \{y'(\phi_1,\nu)\}_i + d & \text{if } R' = R^{(L)}. \end{cases}$$

As constant shifts preserve within-node sums of squared errors, in each of these three scenarios, $\text{DESC}[R', \text{TREE}^0\{y'(\phi_1,\nu)\}] = \text{DESC}[R', \text{TREE}^0\{y'(\phi_2,\nu)\}]$. Thus, $R \in \text{TREE}^0\{y'(\phi_2,\nu)\}$.

Thus, if $R \in \text{TREE}^0\{y'(\phi_1,\nu)\}$, then $R \in \text{TREE}^0\{y'(\phi_2,\nu)\}$. This completes the argument that $\text{TREE}^0\{y'(\phi_1,\nu)\} \subseteq \text{TREE}^0\{y'(\phi_2,\nu)\}$. Swapping the roles of ϕ_1 and ϕ_2 in this argument, we see that $\text{TREE}^0\{y'(\phi_2,\nu)\} \subseteq \text{TREE}^0\{y'(\phi_1,\nu)\}$. This concludes the proof that $\text{TREE}^0\{y'(\phi_1,\nu)\} = \text{TREE}^0\{y'(\phi_2,\nu)\}$.

Because $\text{TREE}^0\{y'(\phi_1,\nu)\} = \text{TREE}^0\{y'(\phi_2,\nu)\}$, it follows that any bottom-up ordering of the regions in $\text{TREE}^0\{y'(\phi_1,\nu)\}$ is also a bottom-up ordering for the regions in $\text{TREE}^0\{y'(\phi_2,\nu)\}$. We next prove by induction that, if we choose a bottom-up ordering $\mathcal{O}$ that places the regions in $\mathcal{R}(\mathcal{B})$ at the end of the ordering, then

$$\text{TREE}_k[\text{TREE}^0\{y'(\phi_1,\nu)\}, y'(\phi_1,\nu), \lambda, \mathcal{O}] = \text{TREE}_k[\text{TREE}^0\{y'(\phi_2,\nu)\}, y'(\phi_2,\nu), \lambda, \mathcal{O}], \quad (41)$$

for $k = 0,\dots,K-L$. It follows immediately from Algorithm A2 and the argument above that $\text{TREE}_0[\text{TREE}^0\{y'(\phi_1,\nu)\}, y'(\phi_1,\nu), \lambda, \mathcal{O}] = \text{TREE}_0[\text{TREE}^0\{y'(\phi_2,\nu)\}, y'(\phi_2,\nu), \lambda, \mathcal{O}]$. Next, suppose that for some $k \in \{1,\dots,K-L\}$,

$$\text{TREE}_{k-1}[\text{TREE}^0\{y'(\phi_1,\nu)\}, y'(\phi_1,\nu), \lambda, \mathcal{O}] = \text{TREE}_{k-1}[\text{TREE}^0\{y'(\phi_2,\nu)\}, y'(\phi_2,\nu), \lambda, \mathcal{O}], \quad (42)$$

and denote this tree with TREE_{k-1} for brevity. We must prove that (41) holds. Let R be the kth region in $\mathcal{O}$ and recall the assumption that the last L regions in $\mathcal{O}$ are $\{R^{(L-1)},\dots,R^{(0)}\}$. Since $k \leq K-L$, this implies that $R \notin \{R^{(L-1)},\dots,R^{(0)}\}$. This means that either

$R \in \text{DESC}\left[R_{sib}^{(l)}, \text{TREE}^0\{y'(\phi_1, \nu)\}\right]$ for $l \in \{1, \ldots, L\}$ or $R \in \text{DESC}\left[R^{(L)}, \text{TREE}^0\{y'(\phi_1, \nu)\}\right]$, meaning that R is a black region in Figure 10(b). From Condition 1,

$$\{y'(\phi_2, \nu)\}_i = \begin{cases} \{y'(\phi_1, \nu)\}_i & \text{if } R \in \text{DESC}\left[R_{sib}^{(l)}, \text{TREE}^0\{y'(\phi_1, \nu)\}\right] \text{ for } l \in \{1, \ldots, L-1\}, \\ \{y'(\phi_1, \nu)\}_i + c & \text{if } R \in \text{DESC}\left[R_{sib}^{(L)}, \text{TREE}^0\{y'(\phi_1, \nu)\}\right], \\ \{y'(\phi_1, \nu)\}_i + d & \text{if } R \in \text{DESC}\left[R^{(L)}, \text{TREE}^0\{y'(\phi_1, \nu)\}\right]. \end{cases}$$

In any of the three cases illustrated in Figure 10, for $g(\cdot)$ defined in (3), $g\{R, \text{TREE}_{k-1}, y'(\phi_1, \nu)\} = g\{R, \text{TREE}_{k-1}, y'(\phi_2, \nu)\}$. Combining this with (42) and Step 2(b) of Algorithm A2 yields (41). This completes the proof by induction.

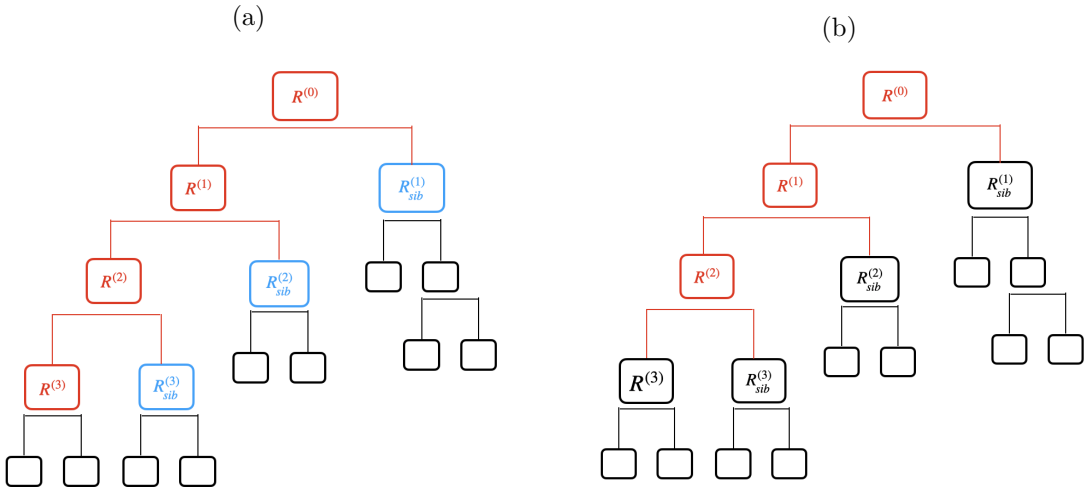

Figure 10: *(a).* An illustration of Case 1 (red), Case 2 (blue), and Case 3 (black) for a region $R \in \text{TREE}^0\{y'(\phi_1, \nu)\}$ in the base case of the proof of Lemma 27, where $\mathcal{R}(\mathcal{B}) = \{R^{(0)}, \ldots, R^{(3)}\}$. *(b.)* The black regions show the possible cases for $R \in \text{TREE}_{k-1}$ in the inductive step of the proof of Lemma 27.

■

Since Lemma 27 guarantees that each $\phi \in S_{grow}(\mathcal{B}, \nu)$ leads to the same $\text{TREE}^0\{y'(\phi, \nu)\}$, we will refer to this tree as TREE^0, will let K be the number of regions in this tree, and will let $\mathcal{O}$ be a bottom-up ordering of these regions that places $R^{(L-1)}, \ldots, R^{(0)}$ in the last L spots. We will further denote $\text{TREE}_{K-L}\{\text{TREE}^0, y'(\phi, \nu), \lambda, \mathcal{O}\}$ for any $\phi \in S_{grow}(\mathcal{B}, \nu)$ as TREE_{K-L}, since Lemma 27 further tells us that this is the same for all $\phi \in S_{grow}(\mathcal{B}, \nu)$. In what follows, we argue that if we let $\text{TREE}(\mathcal{B}, \nu, \lambda) = \text{TREE}_{K-L}$, where $\text{TREE}(\mathcal{B}, \nu, \lambda)$ appears in the statement of Proposition 16, then (21) holds. In other words, we prove that TREE_{K-L}, which always exists and is well-defined, is a valid example of $\text{TREE}(\mathcal{B}, \nu, \lambda)$.

Recall from (18) that $S^{\lambda}(\mathcal{B}, \nu) = \left\{\phi \in S_{grow}(\mathcal{B}, \nu) : R^{(L)} \in \text{TREE}^{\lambda}\{y'(\phi, \nu)\}\right\}$. Lemma 27 says that for $\phi \in S_{grow}(\mathcal{B}, \nu)$, we can rewrite $\text{TREE}^{\lambda}\{y'(\phi, \nu)\}$ as $\text{TREE}_K\{\text{TREE}^0, y'(\phi, \nu), \lambda, \mathcal{O}\}$.

So we can rewrite $S^{\lambda}(\mathcal{B}, \nu)$ as

$$S^{\lambda}(\mathcal{B}, \nu) = \left\{\phi \in S_{grow}(\mathcal{B}, \nu) : R^{(L)} \in \textsc{tree}_K\left\{\textsc{tree}^0, y'(\phi, \nu), \lambda, \mathcal{O}\right\}\right\}. \tag{43}$$

Furthermore, since $R^{(L-1)}, \ldots, R^{(0)}$ (all of which are ancestors of $R^{(L)}$) are the last L nodes in the ordering $\mathcal{O}$, we see that $R^{(L)} \in \textsc{tree}_K\{\textsc{tree}^0, y'(\phi, \nu), \lambda, \mathcal{O}\}$ if and only if no pruning occurs during the last L iterations of Step 2 in Algorithm A2. This means that we can characterize (43) as

$$\left\{\phi \in S_{grow}(\mathcal{B}, \nu) : \textsc{tree}_{K-L}\{\textsc{tree}^0, y'(\phi, \nu), \lambda, \mathcal{O}\} = \textsc{tree}_K\left\{\textsc{tree}^0, y'(\phi, \nu), \lambda, \mathcal{O}\right\}\right\}. \tag{44}$$

Recall that for $k = K - L + 1, \ldots, K$, $R^{(K-k)}$ is the kth region in $\mathcal{O}$, and is an ancestor of $R^{(L)}$. We next argue that we can rewrite (44) as

$$\bigcap_{k=K-L+1}^{K}\left\{\phi \in S_{grow}(\mathcal{B}, \nu) : g\left\{R^{(K-k)}, \textsc{tree}_{K-L}, y'(\phi, \nu)\right\} \geq \lambda\right\}. \tag{45}$$

To begin, suppose that $\phi \in (45)$. As we are talking about a particular ϕ, for $k = 0, \ldots, K$ we will suppress the dependence of $\textsc{tree}_k\left\{\textsc{tree}^0, y'(\phi, \nu), \lambda, \mathcal{O}\right\}$ on its arguments and denote it with $\textsc{tree}_k$. The fact that $\phi \in (45)$ means that $g\left\{R^{(L-1)}, \textsc{tree}_{K-L}, y'(\phi, \nu)\right\} \geq \lambda$, which ensures that no pruning occurs at step $K - L + 1$, which in turn ensures that $\textsc{tree}_{K-L+1} = \textsc{tree}_{K-L}$. Combined with (45), this implies that $g\left\{R^{(L-2)}, \textsc{tree}_{K-L}, y'(\phi, \nu)\right\} = g\left\{R^{(L-2)}, \textsc{tree}_{K-L+1}, y'(\phi, \nu)\right\} \geq \lambda$, which ensures that no pruning occurs at step $K-L+2$, which in turn ensures that $\textsc{tree}_{K-L+2} = \textsc{tree}_{K-L+1} = \textsc{tree}_{K-L}$. Proceeding in this manner, by tracing through the last L iterations of Step 2 of Algorithm A2, we see that ϕ satisfies $\textsc{tree}_K = \textsc{tree}_{K-L}$, and so $\phi \in (44)$.

Next suppose that $\phi \notin (45)$. Let

$$k' = \min_{k \in \{K-L+1, \ldots, K\}}\{k : g\left\{R^{(K-k)}, \textsc{tree}_{K-L}, y'(\phi, \nu)\right\} < \lambda\}.$$

As k' is a minimum, we know that no pruning occurred during steps $K - L + 1, \ldots, k' - 1$, and so $\textsc{tree}_{k'-1}\left\{\textsc{tree}^0(y'(\phi, \nu)), y'(\phi, \nu), \lambda, \mathcal{O}\right\} = \textsc{tree}_{K-L}$. This implies that $g\left\{R^{(K-k')}, \textsc{tree}_{K-L}, y'(\phi, \nu)\right\} < \lambda$ can be rewritten as $g\left(R^{(K-k')}, \textsc{tree}_{k'-1}\left[\textsc{tree}^0\{y'(\phi, \nu)\}, y'(\phi, \nu), \lambda, \mathcal{O}\right], y'(\phi, \nu)\right) < \lambda$. It then follows from Algorithm A2 that pruning occurs at step k', which means that $\textsc{tree}_K$ cannot possibly equal $\textsc{tree}_{K-L}$. Thus, $\phi \notin (44)$.

Thus, $\phi \in (45)$ if and only if $\phi \in (44)$.

Finally, Proposition 16 rewrites (45) with the indexing over k changed to an indexing over l, and plugging in $\textsc{tree}(\mathcal{B}, \nu, \lambda) = \textsc{tree}_{K-L}$. Therefore, $\textsc{tree}_{K-L}$ is a valid example of $\textsc{tree}(\mathcal{B}, \nu, \lambda)$.

To compute $\textsc{tree}(\mathcal{B}, \nu, \lambda)$, we first select an arbitrary $\phi \in S_{grow}(\mathcal{B}, \nu)$. We then apply Algorithm A1 to grow $\textsc{tree}^0\{y'(\phi, \nu)\}$. We create a bottom-up ordering $\mathcal{O}$ of the K nodes in $\textsc{tree}^0\{y'(\phi, \nu)\}$ such that $R^{(L-1)}, \ldots, R^{(0)}$ are at the end. Finally, we apply the first $K - L$ iterations of Algorithm A2 with arguments $\textsc{tree}^0\{y'(\phi, \nu)\}, y'(\phi, \nu), \lambda$, and $\mathcal{O}$ to obtain $\textsc{tree}(\mathcal{B}, \nu, \lambda)$. The worst case computational cost of CART (the combined Algorithm A1 and Algorithm A2) is $O(n^2 p)$.

In the special case that $\mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^{\lambda}(y)$, we have that $\nu^T y \in S_{grow}(\mathcal{B}, \nu)$ because $y = y'(\nu^T y, \nu)$ and $\mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^{\lambda}(y) \subseteq \text{TREE}^{0}(y)$. In this case, suppose that we carry out the process described in the previous paragraph by selecting $\phi = \nu^T y$. As $\mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^{\lambda}(y)$, it is clear from Algorithm A2 that no pruning occurs during the last L iterations of Step 2 in Algorithm A2 applied to arguments $\text{TREE}^{0}(y), y, \lambda$, and $\mathcal{O}$. Therefore, the process from the previous paragraph returns the optimally pruned $\text{TREE}^{\lambda}(y)$. Thus, when $\mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^{\lambda}(y)$, we can simply plug in $\text{TREE}^{\lambda}(y)$, which has already been built, for $\text{TREE}(\mathcal{B}, \nu, \lambda)$ in (21).

D.4 Proof of Proposition 17

We first show that we can express

$$\left\{\phi : g\left\{R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu)\right\} \geq \lambda\right\} \tag{46}$$

as the solution set of a quadratic inequality in ϕ for $l = 0, \ldots, L-1$, where $g(\cdot)$ was defined in (3). Because only the numerator of $g\{R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu)\}$ depends on ϕ, it will be useful to introduce the following concise notation:

$$h(R, \text{TREE}, y) \equiv \sum_{i \in R}(y_i - \bar{y}_R)^2 - \sum_{r \in \text{TERM}(R, \text{TREE})} \sum_{i \in r}(y_i - \bar{y}_r)^2. \tag{47}$$

We begin with the following lemma.

Lemma 28 *Suppose that region R in* TREE *has children $R \cap \chi_{j,s,0}$ and $R \cap \chi_{j,s,1}$. Then, $h(R, \text{TREE}, y) = \text{GAIN}_R(y, j, s) + h(R \cap \chi_{j,s,1}, \text{TREE}, y) + h(R \cap \chi_{j,s,0}, \text{TREE}, y)$, where $\text{GAIN}_R(y, j, s)$ is defined in (2).*

Proof The result follows from adding and subtracting $\sum_{i \in R \cap \chi_{j,s,1}}(y_i - \bar{y}_{R \cap \chi_{j,s,1}})^2$ and $\sum_{i \in R \cap \chi_{j,s,0}}(y_i - \bar{y}_{R \cap \chi_{j,s,0}})^2$ in (47) and noting that $\text{TERM}(R, \text{TREE}) = \text{TERM}(R \cap \chi_{j,s,1}, \text{TREE}) \cup \text{TERM}(R \cap \chi_{j,s,0})$. ■

Recall that $\mathcal{B} = ((j_1, s_1, e_1), \ldots, (j_L, s_L, e_L))$ and that $\mathcal{R}(\mathcal{B}) = \{R^{(0)}, R^{(1)}, \ldots, R^{(L)}\}$. Lemma 29 follows from Lemma 28 and the fact that, due to the form of the vector ν, there are many regions R for which $h\{R, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu)\}$ does not depend on ϕ.

Lemma 29 *For any $\tilde{\phi} \in S_{grow}(\mathcal{B}, \nu)$, we can decompose $h\{R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu)\}$ as*

$$\begin{aligned} h\left\{R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu)\right\} = &\sum_{l'=l+1}^{L} h\left\{R_{sib}^{(l')}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\tilde{\phi}, \nu)\right\} \\ &+ h\left\{R^{(L)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\tilde{\phi}, \nu)\right\} \\ &+ \sum_{l'=l}^{L-1} \text{GAIN}_{R^{(l')}}\{y'(\phi, \nu), j_{l'+1}, s_{l'+1}\}, \end{aligned}$$

where $R_{sib}^{(l)}$ is the sibling of $R^{(l)}$ in TREE$(\mathcal{B}, \nu, \lambda)$.

Proof Repeatedly applying Lemma 28 yields

$$h\left\{R^{(l)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} = \sum_{l'=l+1}^{L} h\left\{R_{sib}^{(l')}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} + h\left\{R^{(L)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} + \sum_{l'=l}^{L-1} \text{GAIN}_{R^{(l')}}\{y'(\phi,\nu), j_{l'+1}, s_{l'+1}\}. \quad (48)$$

For $l' = l+1,\ldots,L-1$, $h\left\{R_{sib}^{(l')}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} = h\left\{R_{sib}^{(l')}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\right\}$ because $R_{sib}^{(l')}$ only contains observations where $[y'(\phi,\nu)]_i = [y'(\tilde{\phi},\nu)]_i$. Similarly, $h\left\{R^{(L)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} = h\left\{R^{(L)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\right\}$ and $h\left\{R_{sib}^{(L)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} = h\left\{R_{sib}^{(L)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\right\}$ because for $i \in R^{(L)}$ and $i \in R_{sib}^{(L)}$, $\{y'(\phi,\nu)\}_i$ and $\{y'(\tilde{\phi},\nu)\}_i$ only differ by a constant shift. Plugging these two facts into (48) completes the proof. ■

We can now write (46) as

$$\begin{aligned}
&\left\{\phi : g\left\{R^{(l)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} \geq \lambda\right\} \\
&= \left\{\phi : h\left\{R^{(l)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\phi,\nu)\right\} \geq \lambda\left[|\,\text{TERM}\{R^{(l)}, \text{TREE}(\mathcal{B},\nu,\lambda)\}| - 1\right]\right\} \\
&= \left\{\phi : \sum_{l'=l}^{L-1} \text{GAIN}_{R^{(l')}}\{y'(\phi,\nu), j_{l'+1}, s_{l'+1}\} \geq \lambda\left[|\,\text{TERM}\{R^{(l)}, \text{TREE}(\mathcal{B},\nu,\lambda)\}| - 1\right] - \right. \\
&\qquad\qquad \left. \sum_{l'=l+1}^{L} h\left\{R_{sib}^{(l')}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\right\} - h\left\{R^{(L)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\right\}\right\} \\
&= \left\{\phi : \sum_{l'=l}^{L-1}\left[a\left\{R^{(l')}, j_{l'+1}, s_{l'+1}\right\}\phi^2 + b\left\{R^{(l')}, j_{l'+1}, s_{l'+1}\right\}\phi + c\left\{R^{(l')}, j_{l'+1}, s_{l'+1}\right\}\right] \geq \gamma_l\right\}, \quad (49)
\end{aligned}$$

where the functions $a(\cdot), b(\cdot)$, and $c(\cdot)$ were defined in (35) in Appendix D.2, and where

$$\gamma_l \equiv \lambda\left[|\,\text{TERM}\{R^{(l)}, \text{TREE}(\mathcal{B},\nu,\lambda)\}| - 1\right] - \sum_{l'=l+1}^{L} h\left\{R_{sib}^{(l')}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\right\} - h\left\{R^{(L)}, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\right\}$$

is a constant that does not depend on ϕ. The first equality simply applies the definitions of $h(\cdot)$ and $g(\cdot)$. The second equality follows from Lemma 29 and moving terms that do not depend on ϕ to the right-hand-side. The third equality follows from plugging in notation from Appendix D.2 and defining the constant γ_l for convenience. Thus, (46) is quadratic inequality in ϕ. We now just need to argue that its coefficients can be obtained efficiently.

We need to compute the coefficients in (49) for $l = 0,\ldots,L-1$. The quantities $a\{R^{(l')}, j_{l'}, s_{l'+1}\}$, $b\{R^{(l')}, j_{l'+1}, s_{l'+1}\}$, and $c\{R^{(l')}, j_{l'+1}, s_{l'+1}\}$ for $l' = 0,\ldots,L-1$ were already computed while computing $S_{grow}(\mathcal{B},\nu)$. To get the coefficients for the left hand side of (49) for each $l = 0,\ldots,L-1$, we simply need to compute L partial sums of these quantities, which takes $O(L)$ operations. As we are assuming that we have access to $\text{TREE}(\mathcal{B},\nu,\lambda)$, computing $h\{R, \text{TREE}(\mathcal{B},\nu,\lambda), y'(\tilde{\phi},\nu)\}$ requires $O(n)$ operations. Therefore, computing γ_0

takes $O(nL)$ operations. By storing partial sums during the computation of γ_0, we can subsequently obtain γ_l for $l = 1, \ldots, L-1$ in constant time.

We have now seen that we can obtain the coefficients needed to express (46) as a quadratic function of ϕ for $l = 0, \ldots, L-1$ in $O(nL)$ total operations. Once we have these quantities, we can compute each set of the form (46) in constant time using the quadratic equation.

It remains to compute

$$S^{\lambda}(\mathcal{B}, \nu) = S_{grow}(\mathcal{B}, \nu) \bigcap \left[\bigcap_{l=0}^{L-1} \left\{ \phi : g\left\{ R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu) \right\} \geq \lambda \right\} \right]. \tag{50}$$

Recall from Proposition 14 that $S_{grow}(\mathcal{B}, \nu)$ is the intersection of $O(npL)$ quadratic sets. Thus, in the worst case, $S_{grow}(\mathcal{B}, \nu)$ has $O(npL)$ disjoint components, and so this final intersection involves $O(npL)$ components. Thus, we can compute $S^{\lambda}(\mathcal{B}, \nu)$ in $O\{npL \times log(npL)\}$ operations (Bourgon, 2009).

The following lemma explains why, similar to Proposition 15, computation time can be reduced in the special case where $\nu = \nu_{sib}$ and $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$.

Lemma 30 *When $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$, the set $\{\phi : g\{R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu_{sib})\} \geq \lambda\}$ has the form $(-\infty, a_l) \cup (b_l, \infty)$, where $a_l \leq 0 \leq b_l$. Therefore, we can compute $\bigcap_{l=0}^{L-1} \left\{ \phi : g\left\{ R^{(l)}, \text{TREE}(\mathcal{B}, \nu, \lambda), y'(\phi, \nu) \right\} \geq \lambda \right\}$ as*

$$\bigcap_{l=0}^{L-1} (-\infty, a_l) \cup (b_l, \infty) = (-\infty, \min_{0 \leq l \leq L-1} a_l) \cup (\max_{0 \leq l \leq L-1} b_l, \infty)$$

in $O(L)$ operations. Furthermore, we can compute (50) in constant time.

Proof

As $\mathcal{R}(\mathcal{B}) \subseteq \text{TREE}^{\lambda}(y)$, we can let $\tilde{\phi} \in S_{grow}(\mathcal{B}, \nu)$ from Lemma 29 be $\nu^T y$ such that $y'(\tilde{\phi}, \nu) = y$. We can then apply Lemma 22 to note that

$$\sum_{l'=l}^{L-1} \text{GAIN}_{R^{(l')}}\{y'(\phi, \nu_{sib}), j_{l'+1}, s_{l'+1}\} = \left[\sum_{l'=l}^{L-2} \text{GAIN}_{R^{(l')}}\{y, j_{l'+1}, s_{l'+1}\} \right] + \text{GAIN}_{R^{(L-1)}}\{y'(\phi, \nu_{sib}), j_L, s_L\}.$$

Thus, when $\nu = \nu_{sib}$ and $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^{\lambda}(y)\}$, we can rewrite (49) with all of the terms corresponding to $\text{GAIN}_{R^{(l')}}\{y'(\phi, \nu_{sib}), j_{l'+1}, s_{l'+1}\}$ for $l' = l+1, \ldots, L-2$ moved into the constant on the right-hand-side. This lets us rewrite (49) as $\{\phi : Gain_{R^{(L-1)}}\{y'(\phi, \nu_{sib}), j_L, s_L\} \geq \tilde{\gamma}_l\}$, where $\tilde{\gamma}_l$ is an updated constant that does not depend on ϕ.

To prove that $\{\phi : Gain_{R^{(L-1)}}\{j_L, s_L, y'(\phi, \nu_{sib})\} \geq \tilde{\gamma}_l\}$ has the form $(-\infty, a_l) \cup (b_l, \infty)$ for $a_l \leq 0 \leq b_l$, first recall from Lemma 23 that $Gain_{R^{(L-1)}}\{j_L, s_L, y'(\phi, \nu_{sib})\}$ is a quadratic function of ϕ. It then suffices to show that this quadratic has a non-negative second derivative and achieves its minimum when $\phi = 0$. The second derivative of this quadratic is $a\left\{R^{(L-1)}, j_L, s_L\right\} = ||\nu_{sib}||_2^{-4} \nu_{sib}^T M_{R^{(L-1)}, j_L, s_L} \nu_{sib}$, which is non-negative by Lemma 23. From Lemma 23, $Gain_{R^{(L-1)}}\{j_L, s_L, y'(\phi, \nu_{sib})\}$ is non-negative. It equals 0 when $\phi = 0$, because when $\phi = 0$ then $\bar{y}_{R^{(L-1)} \cap \chi_{j_L, s_L, 1}} = \bar{y}_{R^{(L-1)} \cap \chi_{j_L, s_L, 0}}$.

Intersecting L sets of the form $(-\infty, a_l) \cup (b_l, \infty)$ for $a_l \leq 0 \leq b_l$ only takes $O(L)$ operations, because we simply need to identify the minimum a_l and maximum b_l. Finally, Lemma 26 ensures that $S_{grow}(\mathcal{B}, \nu_{sib})$ has at most two disjoint intervals, and so the final intersection with $S_{grow}(\mathcal{B}, \nu_{sib})$ takes only $O(1)$ operations. ■

D.5 Proof of Proposition 18

Let $\mathcal{B} = \text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}$, let $\mathcal{R}(\mathcal{B}) = \{R^{(0)}, \dots, R^{(L)}\}$, and let $\nu = \nu_{sib}$ (6). Applying the expression given for $\{y'(\phi, \nu_{sib})\}_i$ in Section 3.3 (which follows from algebra), we immediately see that Condition 1 holds with $R_A = R^{(L)}$ and $R_B = R^{(L-1)} \cap \chi_{j_L, s_L, 1-e_L}$.

Let $\mathcal{B} = \pi\left[\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}\right]$ and let $\nu = \nu_{reg}$ (14). Note that $\pi\left[\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}\right]$ induces the same region $R^{(L)}$ as the unpermuted $\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}$. Regardless of the permutation, the induced $R^{(L)}$ is equal to R_A. Applying the expression for $\{y'(\phi, \nu_{reg})\}_i$ given in Section 3.3, Condition 1 holds with constant $c_2 = 0$.

Appendix E. Proofs for Section 4.3

E.1 Proof of Proposition 19

As stated in Proposition 19, let

$$p^Q_{reg}(y) = pr_{H_0}\left\{|\nu_{reg}^\mathrm{T} Y - c| \geq |\nu_{reg}^\mathrm{T} y - c| \mid \bigcup_{\pi \in Q} \mathcal{R}(\pi[\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}]) \subseteq \text{TREE}^\lambda(Y), \mathcal{P}^\perp_{\nu_{reg}} Y = \mathcal{P}^\perp_{\nu_{reg}} y\right\}.$$

First, we will show that the test based on $p^Q_{reg}(y)$ controls the selective Type 1 error rate, defined in (4); this is a special case of Proposition 3 from Fithian et al. (2014).

Define $\mathcal{E}_1 = \{Y : R_A \in \text{TREE}^\lambda(Y)\}$, $\mathcal{E}_2 = \{Y : \mathcal{P}^\perp_{\nu_{reg}} Y = \mathcal{P}^\perp_{\nu_{reg}} y\}$, and $\mathcal{E}_3 = \left\{Y : \bigcup_{\pi \in Q} \mathcal{R}\left(\pi\left[\text{BRANCH}\left\{R_A, \text{TREE}^\lambda(y)\right\}\right]\right) \subseteq \text{TREE}^\lambda(Y)\right\}$. Recall that the test of $H_0 : \nu_{reg}^\mathrm{T} \mu = c$ based on $p^Q_{reg}(y)$ controls the selective Type 1 error rate if, for all $\alpha \in [0, 1]$, $pr_{H_0}\{p^Q_{reg}(Y) \leq \alpha \mid \mathcal{E}_1\} \leq \alpha$. By construction,

$$pr_{H_0}\left\{p^Q_{reg}(Y) \leq \alpha \mid \mathcal{E}_2 \cap \mathcal{E}_3\right\} = E\left[1_{\{pr_{H_0}(|\nu_{reg}^\mathrm{T} Y - c| \geq |\nu_{reg}^\mathrm{T} y - c| \mid \mathcal{E}_2 \cap \mathcal{E}_3) \leq \alpha\}} \mid \mathcal{E}_2 \cap \mathcal{E}_3\right] = \alpha.$$

Let $\psi^Q_{R_A} = 1_{\{pr_{H_0}(|\nu_{reg}^\mathrm{T} Y - c| \geq |\nu_{reg}^\mathrm{T} y - c| \mid \mathcal{E}_2 \cap \mathcal{E}_3) \leq \alpha\}}$. An argument similar to that of Lemma 13 indicates that $\mathcal{E}_3 \subseteq \mathcal{E}_1$. The law of total expectation then yields

$$E(\psi^Q_{R_A} \mid \mathcal{E}_1) = E\left\{E(\psi^Q_{R_A} \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3) \mid \mathcal{E}_1\right\} = E\left\{E(\psi^Q_{R_A} \mid \mathcal{E}_2 \cap \mathcal{E}_3) \mid \mathcal{E}_1\right\} = E(\alpha \mid \mathcal{E}_1) = \alpha.$$

Thus, the test based on $p^Q_{reg}(y)$ controls the selective Type 1 error rate. We omit the proof that $p^Q_{reg}(y)$ can be computed as

$$p^Q_{reg}(y) = pr_{H_0}\left\{|\phi - c| \geq |\nu_{reg}^\mathrm{T} y - c| \mid \phi \in \bigcup_{\pi \in Q} S^\lambda\left(\pi[\text{BRANCH}\{R_A, \text{TREE}^\lambda(y)\}], \nu_{reg}\right)\right\}$$

for $\phi \sim N(c, \|\nu_{reg}\|_2^2 \sigma^2)$, as the proof is similar to the proof of Theorem 6.

Appendix F. Effect of Using the Computationally Efficient Alternative in Section 4.3

In this section, we investigate the effect of using $p^{\mathcal{I}}_{reg}(y)$ from (22) rather than $p_{reg}(y)$ from (15) on power. We also investigate the effect of using (23) rather than (17) on the width of confidence intervals for $\nu^{\mathrm{T}}_{reg}\mu$.

We generate data as described in Section 5.1, but for simplicity we restrict our attention to the case where $a = 1$ (corresponding to the center panel of Figure 3).

For each tree that we build, we consider (a) testing $H_0 : \nu^{\mathrm{T}}_{reg}\mu = 0$ and (b) constructing a confidence interval for $\nu^{\mathrm{T}}_{reg}\mu$, for each region appearing at the third level of the tree. For each test, we compare the test that uses the full conditioning set (i.e. that uses $p_{reg}(y)$ from (15)) to the test that uses the identity permutation only (i.e. that uses $p^{\mathcal{I}}_{reg}(y)$ from (22)). For each interval, we compare the method that uses the full conditioning set (i.e. (17)) to the method that uses the identity permutation only (i.e. (23)). The results are displayed in Figure 11.

The left panel of Figure 11 shows that the power loss resulting from using (22) instead of (15) is negligible. In fact, we need to zoom in on the left panel, as shown in the center panel, to see any separation between the power curves. We see in the center panel that power is lower when (22) is used, though we emphasize that the differences in power are extremely small.

We see in the right panel of Figure 11 that the computationally efficient conditioning set has a more noticeable impact on the median width of our confidence intervals. As expected, the confidence intervals are narrower when we use the full conditioning set (i.e. (17)). However, this difference is most noticeable when b is small. When b is small, in which case the confidence intervals are wide even when the full conditioning set is used. Thus, the amount of precision lost overall by constructing confidence intervals using the identity permutation only (i.e. (23)) is not of practical importance.

Based on these results, we recommend using the identity permutation in practice, because it is the most computationally efficient choice *and* does not meaningfully reduce power or precision compared to the full conditioning set.

Appendix G. Robustness to Non-Normality

In this section, we explore the performance of the selective Z-test under a global null when the normality assumption on Y is violated. For four choices of cumulative distribution function F, we generate $Y_i \overset{\text{i.i.d.}}{\sim} F$ such that all observations have the same expected value. We set $n = 200, p = 10$, and we grow trees to a maximum depth of 3. We plug in $(n-1)^{-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$ as an estimate of σ^2, as it does not make sense to assume known variance for distributions with a mean-variance relationship. Figure 12 displays quantile-quantile plots of the p-values for testing $H_0 : \nu^{\mathrm{T}}_{sib}\mu = 0$, using the test in (8).

Figure 12 shows that despite the fact that our proposed selective inference framework was derived under a normality assumption, it yields approximately uniformly distributed p-values for Poisson(10), Bernoulli(0.5), and Gamma(1,10) data. We suspect that this is because $\mathcal{P}^{\perp}_{\nu}Y$ is approximately independent of $\nu^{\mathrm{T}}Y$ for these distributions (see the proof of Theorem 6 in Appendix B). In the case of Bernoulli(0.1) data, which represents a par-

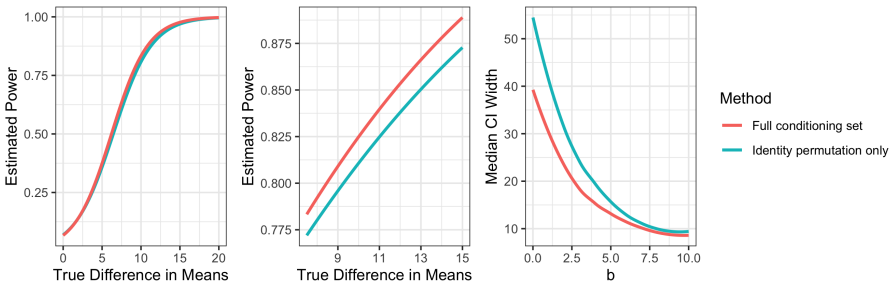

Figure 11: Simulation results comparing inference based on the full conditioning set to inference based on the identity permutation only (see Section 4.3). The left panel shows power curves. The center panel zooms in on one section of the left panel. The right panel shows median widths of confidence intervals.

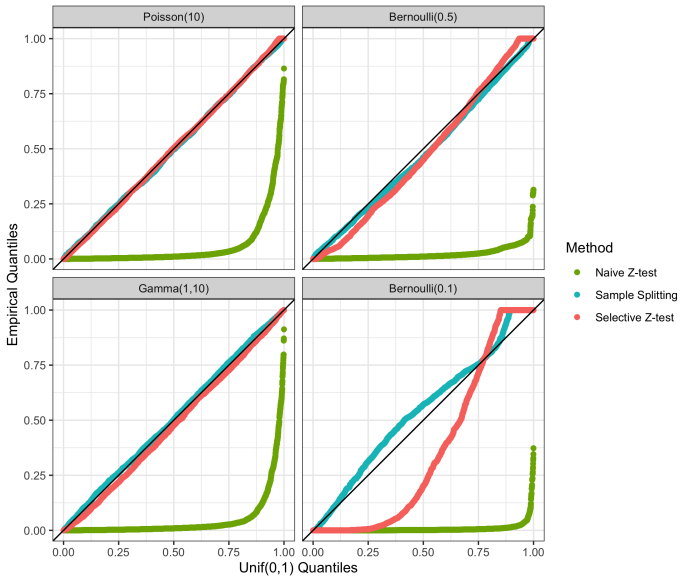

Figure 12: Quantile-quantile plots of the p-values for testing $H_0 : \nu_{sib}^{\mathrm{T}}\mu = 0$ under a global null. A naive Z-test (green), sample splitting (blue), and selective Z-test (pink) were performed; see Section 5.2.

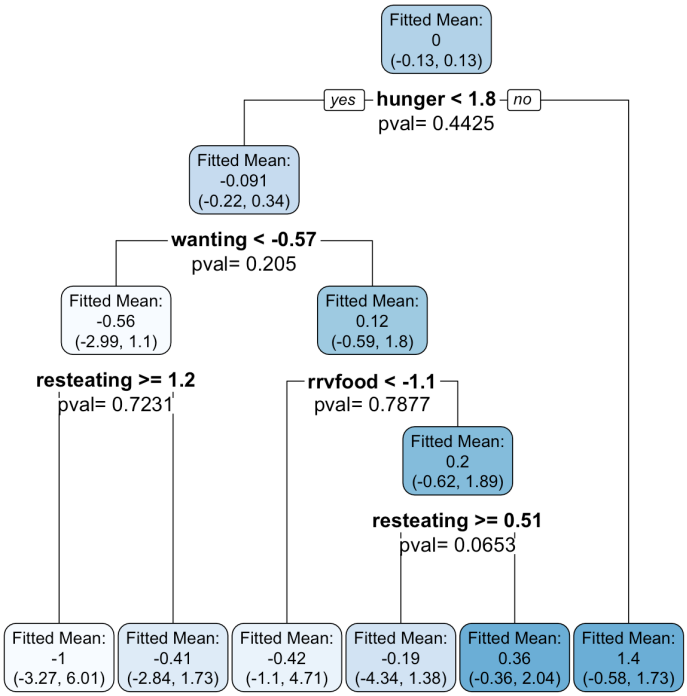

Figure 13: A CART tree fit to the Box Lunch Study data. Each split has been labeled with a p-value (8), and each region has been labeled with a confidence interval (23). Inference is carried out by plugging in $\hat{\sigma}_{\text{cons}}$, from Section 5.7, as an estimate of σ.

ticularly extreme violation of the normality assumption, the p-values from our selective Z-test are not uniformly distributed. As mentioned in Section 7, future work could involve characterizing conditions for F under which our selective Z-tests will approximately control the selective Type 1 error.

Appendix H. Alternate Box Lunch Study Analysis

Figure 13 is the same as the left panel of Figure 9, but the selective Z-inference is carried out with $\hat{\sigma}_{\text{cons}}$, from Section 5.7, rather than $\hat{\sigma}_{\text{SSE}}$. The takeaways presented in Section 6 do not change.

References

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

PK Bhattacharya. Some aspects of change-point analysis. *Lecture Notes-Monograph Series*, pages 28–56, 1994.

Richard Bourgon. *Overview of the intervals package*, 2009. R Vignette, URL `https://cran.r-project.org/web/packages/intervals/vignettes/intervals_overview.pdf`.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC Press, 1984.

Shuxiao Chen and Jacob Bien. Valid inference corrected for outlier removal. *Journal of Computational and Graphical Statistics*, 29(2):323–334, 2020.

Yiqun T Chen and Daniela M Witten. Selective inference for k-means clustering. *arXiv preprint arXiv:2203.15267*, 2022.

William Fithian, Dennis Sun, and Jonathan Taylor. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*, 2014.

Lucy L Gao, Jacob Bien, and Daniela Witten. Selective inference for hierarchical clustering. *arXiv preprint arXiv:2012.02936*, 2020.

Torsten Hothorn and Achim Zeileis. partykit: A modular toolkit for recursive partytioning in R. *The Journal of Machine Learning Research*, 16(1):3905–3909, 2015.

Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15 (3):651–674, 2006.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, 1985.

Sangwon Hyun, Kevin Z Lin, Max G'Sell, and Ryan J Tibshirani. Post-selection inference for changepoint detection algorithms with application to copy number variation data. *Biometrics*, pages 1–13, 2021.

Sean Jewell, Paul Fearnhead, and Daniela Witten. Testing for a change in mean after changepoint detection. *Journal of the Royal Statistical Society, Series B*, 2022.

Danijel Kivaranovic and Hannes Leeb. On the length of post-model-selection confidence intervals conditional on polyhedral constraints. *Journal of the American Statistical Association*, 116(534):845–857, 2021.

Jason D Lee, Dennis L Sun, Yuekai Sun, Jonathan E Taylor, et al. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927, 2016.

Keli Liu, Jelena Markovic, and Robert Tibshirani. More powerful post-selection inference, with application to the lasso. *arXiv preprint arXiv:1801.09037*, 2018.

Wei-Yin Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348, 2014.

Wei-Yin Loh, Haoda Fu, Michael Man, Victoria Champion, and Menggang Yu. Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables. *Statistics in Medicine*, 35(26):4837–4855, 2016.

Wei-Yin Loh, Michael Man, and Shuaicheng Wang. Subgroups from regression trees with adjustment for prognostic effects and postselection inference. *Statistics in Medicine*, 38 (4):545–557, 2019.

Brian D Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996.

Mark Robert Segal. Regression trees for censored data. *Biometrics*, 44(1):35–47, 1988.

Jonathan Taylor and Robert J Tibshirani. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634, 2015.

Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2019. R package version 4.1-15, available on CRAN.

Xiaoying Tian and Jonathan Taylor. Asymptotics of selective inference. *Scandinavian Journal of Statistics*, 44(2):480–499, 2017.

Xiaoying Tian and Jonathan Taylor. Selective inference with a randomized response. *The Annals of Statistics*, 46(2):679–710, 2018.

Ryan J Tibshirani, Jonathan Taylor, Richard Lockhart, and Robert Tibshirani. Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620, 2016.

Ryan J Tibshirani, Alessandro Rinaldo, Rob Tibshirani, and Larry Wasserman. Uniform asymptotic inference and the bootstrap after model selection. *The Annals of Statistics*, 46(3):1255–1287, 2018.

Ashwini Venkatasubramaniam and Julian Wolfson. *visTree: Visualization of Subgroups for Decision Trees*, 2018. R package version 0.8.1, available on CRAN.

Ashwini Venkatasubramaniam, Julian Wolfson, Nathan Mitchell, Timothy Barnes, Meghan JaKa, and Simone French. Decision trees in epidemiological research. *Emerging Themes in Epidemiology*, 14(1):11, 2017.

Stefan Wager and Guenther Walther. Adaptive concentration of regression trees, with application to random forests. *arXiv preprint arXiv:1503.06388*, 2015.