Artificial Intelligence Review _####################_
https://doi.org/10.1007/s10462-026-11541-6

Super greedy trees

Hemant Ishwaran[1]

Received: 11 November 2025 / Accepted: 13 March 2026

Abstract
We introduce Super Greedy Trees (SGTs), a decision-tree framework that extends CART by constructing tree splits from lasso-penalized parametric models. At each tree node, a model fitted to the local data induces an adaptive multivariate geometric cut (linear or curved) selected to greedily reduce empirical risk. This yields richer partitions than axis-parallel CART while keeping each split easy to inspect through sparse local structure. In simulated and real-world regression studies, SGTs and an ensemble extension (Super Greedy Forests, SGFs) perform well relative to CART, oblique trees, random forests, and gradient boosted trees, especially when the underlying response surface is complex. In a treadmill ECG and clinical-data case study, SGFs identify sparse combinations of signals associated with long-term survival. The SGT framework thus provides a flexible and theoretically sound approach to tree-based learning.

Keywords Empirical risk · Lasso · Multivariate cuts · Parametric models · Partitions

1 Introduction

Classification and Regression Trees (CART) (Breiman et al. 1984) are widely used for prediction and decision rules across many fields. They are nonparametric, handle mixed types of predictors, and often perform well with limited tuning. Tree-based learners also serve as fundamental building blocks for many successful ensemble methods, including random forests (Breiman 2001), Bayesian additive regression trees (BART) (Chipman et al. 2010), and gradient boosted trees (GBT) (Freund and Schapire 1996; Friedman 2001). A widely used GBT implementation is XGBoost (Chen and Guestrin 2016), known for strong performance in machine learning benchmarks and Kaggle competitions. CART grows a tree by repeatedly choosing, at each node, a split (cut) that divides the observations into two child nodes and maximizes the decrease in an impurity criterion. In regression this is typically the

✉ Hemant Ishwaran
hishwaran@miami.edu

[1] Division of Biostatistics, Miller School of Medicine, University of Miami, Miami, USA

within-node sum of squares; in classification it is often the Gini index. These criteria correspond to greedy reductions in the average training loss (empirical risk), so CART can be viewed as empirical risk minimization carried out one split at a time.

Despite its success, CART restricts the split search to univariate, axis-parallel rules of the form $x_l \leq s$. Such splits are easy to understand, but they yield piecewise-constant prediction surfaces and partition boundaries that are parallel to the coordinate axes. When the signal depends on multivariate directions or curved boundaries, CART often requires many levels to approximate the structure, producing deep trees. Deep trees can have high variance and may be unstable to small perturbations in the data (Breiman 1996) (see also the survey by Loh (2014)).

Many extensions enlarge the split family. Oblique trees replace axis-parallel splits by hyperplanes defined through linear combinations of features (Murthy et al. 1994; Heath et al. 1993). Projection pursuit trees and forests choose splits along data-driven linear projections (Lee et al. 2013; da Silva et al. 2021). Optimization-based approaches learn sparse oblique splits by directly minimizing a training objective (Carreira-Perpiñán and Tavallali 2018; Zharmagambetov and Carreira-Perpiñán 2020); related forest methods include sparse projection oblique random forests (Tomita et al. 2020). Other lines of work enrich trees by fitting local parametric structure [e.g., local linear forests (Friedberg et al. 2020)] or by learning feature-space rotations [rotation forests; Rodriguez et al. 2006]. While effective, these approaches typically commit to a single split geometry, most commonly hyperplanes, and therefore do not, within one greedy risk-reduction procedure, switch between qualitatively different cut types when the data call for it.

We address this gap by introducing *Super Greedy Trees* (SGTs). At each node, SGTs fit a lasso-penalized parametric model (Tibshirani 1996; Friedman et al. 2010) on the observations that fall in that node and use the fitted model to propose candidate multivariate geometric cuts. Depending on the model class, the induced cut can be linear (a hyperplane) or curved, and the final split is chosen following a greedy empirical-risk reduction principle like CART. We call the method "super greedy" because each step searches over a richer, model-derived collection of splits than the coordinate-threshold search of CART. Moreover, when compared to model trees (Quinlan 1992; Frank et al. 1998), which use parametric models primarily for terminal-node predictions, SGTs use them to *define* the splits.

This paper is organized as follows. Section 2 reviews related work on decision trees and multivariate splitting, emphasizing how split geometry and split-search procedures determine the induced partition class, stability, and empirical risk. Section 3 introduces SGTs, presents the lasso-based splitting rule, and develops theoretical properties, including that SGTs yield strictly richer partition spaces than conventional trees. Section 4 reports empirical results on simulated and real datasets. Section 5 presents a treadmill exercise ECG case study for predicting long-term mortality, illustrating how sparse local model fits can highlight interactions and subgroups. Section 6 discusses computational considerations and future directions.

2 Split families, split-search rules, and prior methods

Tree algorithms differ in two closely related ways. First, they restrict the *split family*, meaning the geometric form of admissible cuts. Second, they specify a *split search rule* that chooses, at each node, a particular cut from that family, typically by maximizing the decrease in an impurity or loss criterion. We briefly review these choices for CART and for multivariate splitting methods, and we summarize related work that motivates the SGT framework.

We begin by recalling the split rule used by CART. A node corresponds to a region (cell) $A \subset \mathbb{R}^p$. A univariate split on coordinate x_l at threshold s produces two child regions

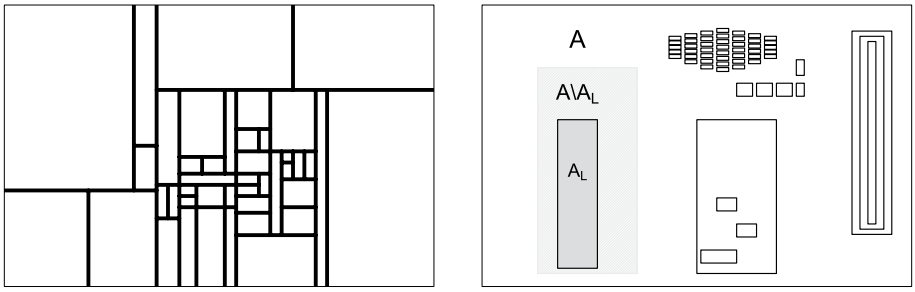
$$A_L = \{\mathrm{x} \in A : x_l \le s\}, \qquad A_R = \{\mathrm{x} \in A : x_l > s\},$$

where $\mathrm{x} = (x_1, \ldots, x_p)^T \in \mathbb{R}^p$ and $l \in \{1, \ldots, p\}$. Each split is a coordinate-parallel hyperplane orthogonal to the x_l-axis, and repeated application partitions $\mathbb{R}^p$ into a collection of axis-parallel hyperrectangles (Figure 1). When the underlying decision boundary is nonlinear or driven by interactions, such rectangular partitions may require deep trees to approximate the target structure, which can increase variance and raise the risk of overfitting (Breiman 1996). Recent work has also sharpened our understanding of the statistical behavior of CART-style trees in large-scale prediction settings, clarifying approximation and estimation tradeoffs and identifying settings in which trees remain competitive (Klusowski and Tian 2024).

Beyond geometric restrictions, impurity-based split selection can exhibit variable-selection biases when predictors differ in scale, missingness patterns, or the number of candidate split points available for a given variable. Several algorithms seek to mitigate these effects using bias-corrected or statistically motivated split-selection rules, including QUEST (Loh and Shih 1997), GUIDE (Loh 2002), and the conditional inference framework (Hothorn et al. 2006).

A richer family of partitions arises from multivariate cuts, where splits occur along hyperplanes of the form

$$\beta_1 x_1 + \cdots + \beta_p x_p \le \alpha_0. \tag{1}$$



(a) *CART partition.* (b) *Greedy geometric cuts.*

Fig. 1 Comparison of axis-parallel CART partitions with richer greedy geometric cuts

Given a node region A, such a cut produces $A_L = A \cap \{\mathrm{x} : \beta^T \mathrm{x} \leq \alpha_0\}$ and $A_R = A \cap \{\mathrm{x} : \beta^T \mathrm{x} > \alpha_0\}$. Recursive application yields leaf cells that are (typically) polyhedra described as intersections of half-spaces. These oblique splits can represent a broader class of decision boundaries than coordinate-threshold cuts. Even so, when the target boundary is curved or otherwise structured, hyperplane partitions may still require many splits unless the split geometry is chosen in a way that reflects local structure in the data.

Classical theory makes clear that both the split family and the split search rule can matter for statistical validity. Devroye et al.(1996,Ch. 20.8) give a "checkerboard" example showing that a natural greedy tree rule based on coordinate-threshold splits can be inconsistent in certain classification settings (see also Ferreira (2022) for discussion of related issues in random forests). They also show that Bayes consistency can be recovered under mild conditions by using appropriately designed greedy procedures and by controlling tree growth. More broadly, this line of work highlights that greedy procedures targeting empirical risk reduction can yield consistent classifiers, provided complexity is suitably regulated.

Related modern theory establishes consistency for oblique-tree and ensemble constructions under mild conditions (Zhan et al. 2025), and provides unifying consistency results for random-forest-type algorithms under probabilistic impurity-decrease conditions that accommodate oblique regression tree methods (Blum et al. 2024). Complementing this theoretical picture, recent empirical work identifies settings where the standard CART split criterion can miss interaction structure, motivating alternative split search and partitioning schemes (Blum et al. 2025).

The idea that multivariate cuts can enhance tree-based learning has been explored extensively (see Loh (2014) for a broad review). Some work considers rectangular or nested-rectangle splits to address limitations of simple univariate rules (Salzberg 1991; Wettschereck and Dietterich 1995; Frank and Witten 1998). However, much of the literature focuses on oblique trees with splits of the form (1). Cattaneo et al. (2024) showed that oblique regression trees can, in principle, match the accuracy of neural networks under similar regression models. Earlier studies by Heath et al. (1993), Murthy et al. (1994), and Brodley and Utgoff (1995) investigated induction using hyperplane splits. Projection pursuit classification trees (Lee et al. 2013) and projection pursuit forests (da Silva et al. 2021) provide a related approach by selecting splits through data-adaptive linear projections.

A complementary line of work improves the computational search for high-quality multivariate splits. Bertsimas and Dunn (2017) proposed optimal hyperplane trees using mixed-integer optimization. Related optimization-based approaches include scalable MIP formulations for optimal multivariate trees (Zhu et al. 2020), near-optimal nonlinear regression trees (Bertsimas et al. 2021), and MaxSAT-based methods for learning globally optimal oblique trees (Avellaneda 2025). Other procedures include tree alternating optimization for learning sparse oblique trees and forests (Carreira-Perpiñán and Tavallali 2018; Zharmagambetov and Carreira-Perpiñán 2020), and SVM-based oblique regression trees such as TORS (Carta and Frigau 2025). Oblique tree ensembles were studied by Menze et al. (2011) and Tomita et al. (2020). Breiman (2001) introduced Forest-RC, which forms splits from linear combinations of inputs. Related developments include rotation forests (Rodriguez et al. 2006; Blaser and Fryzlewicz 2016) and canonical correlation forests (Rainforth and Wood 2015), which incorporate feature correlations through rotated or CCA-based hyperplane splits. Oblique splits have also been incorporated into Bayesian tree models, including recent oblique BART constructions (Nguyen et al. 2025).

Model-oriented approaches, such as local linear forests (Friedberg et al. 2020) and model trees (Quinlan 1992; Frank et al. 1998), instead fit parametric models in terminal nodes to capture local structure. Related methods include logistic regression trees such as LOTUS (Chan and Loh 2004), logistic model trees (Landwehr et al. 2005), and model-based recursive partitioning (Zeileis et al. 2008), which combines parametric models with recursive partitioning via statistically guided splitting. These procedures typically retain univariate split geometry while improving within-node modeling.

Overall, current methods improve tree learners by adding multivariate splits or by using richer terminal nodel models, but most fix the split family in advance, often as hyperplanes, rather than selecting among qualitatively different cut geometries. Our approach departs from this by letting split geometry and prediction be determined locally from the data, which allows more targeted partitioning combined with greedy empirical risk reduction. This forms the basis of the Super Greedy Tree (SGT) framework developed in the next section.

3 Construction and greedy risk reduction

Having reviewed split families and split-search rules, we now describe how Super Greedy Trees (SGTs) are built. SGTs extend classical decision trees in two ways. First, node splits are defined by multivariate *geometric cuts* induced by a structured class of sparse parametric models. Second, tree growth follows a *best-split-first* (BSF) strategy. At each step, the algorithm evaluates every current cell and carries out the split that yields the largest decrease in empirical risk. Standard recursive partitioning typically expands one node at a time according to a fixed traversal order, such as depth-first or breadth-first.

Although SGTs can be applied more broadly, we focus on nonparametric regression. Let $Y \in \mathcal{Y}$ be a scalar response and $\mathrm{X} \in \mathcal{X} \subseteq \mathbb{R}^p$ a (continuous) feature vector. The training data are $(\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, assumed to follow

$$y_i = \psi(\mathrm{x}_i) + \varepsilon_i, \qquad \psi(\mathrm{x}) = \mathbb{E}[Y \mid \mathrm{X} = \mathrm{x}],$$

where ψ is the regression function. For any region $A \subseteq \mathcal{X}$, let $M(A) = \sum_{i=1}^{n} 1_{\{\mathrm{x}_i \in A\}}$ denote the number of training observations falling in A.

We build a sequence of partitions $\{\Pi_k\}_{k \geq 0}$, where $\Pi_0 = \{\mathcal{X}\}$ and, after k splits, Π_k contains $k+1$ cells. At step k, the algorithm selects the cell $A^* \in \Pi_{k-1}$ with maximal empirical risk reduction and replaces it with two child cells A_L^* and A_R^*, yielding the refined partition $\Pi_k = (\Pi_{k-1} \setminus \{A^*\}) \cup \{A_L^*, A_R^*\}$. After K splits, the resulting piecewise estimator has the form

$$\hat{\psi}_{\Pi_K}(\mathrm{x}) \;=\; \sum_{A \in \Pi_K} \hat{\psi}_A(\mathrm{x})\, 1_{\{\mathrm{x} \in A\}},$$

where $\hat{\psi}_A$ is a local model fitted using the observations falling in cell A. The entire procedure is summarized in Algorithm 1. In the regression setting, empirical risk refers to the average training loss, and in particular the mean squared error when squared loss is used (which will be used in this paper). The quantity $R(A)$ appearing in Algorithm 1 denotes the

decrease in empirical risk obtained by splitting A using its best permissible cut and refitting the two child models.

Require: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$; number of splits K; minimum node size $m_{\min}$; permissible cut class $\mathcal{A}$.

Initialization

Start with $\pi_0 = \{\mathcal{X}\}$.

BSF growth

for $k = 1$ to K **do**
  for each terminal cell $A \in \pi_{k-1}$ **do**
    Fit local model $\hat{\psi}_A$ using data with $\mathbf{x}_i \in A$.
    Find the best permissible cut (A_L, A_R) induced by $\hat{\psi}_A$ (subject to $M(A_L) \geq m_{\min}$ and $M(A_R) \geq m_{\min}$).
    Compute the empirical risk reduction $R(A)$ for this split.
  end for
  Choose $A^* = \arg\max_{A \in \pi_{k-1}} R(A)$.
  Split A^* into child cells A_L^* and A_R^* and update $\pi_k = (\pi_{k-1} \setminus \{A^*\}) \cup \{A_L^*, A_R^*\}$.
end for

Output

Return π_K and $\hat{\psi}_{\pi_K}(\mathbf{x}) = \sum_{A \in \pi_K} \hat{\psi}_A(\mathbf{x})\, 1_{\{\mathbf{x} \in A\}}$.

Algorithm 1 Super Greedy Tree (SGT)

Remark 1 At any stage of the algorithm, each current cell A carries a fitted local model $\hat{\psi}_A$ that may vary over $\mathrm{x} \in A$ (in this paper, $\hat{\psi}_A$ is a lasso fit within a parametric regression family). When a cell A^* is split into child cells A_L^* and A_R^*, the contribution $\hat{\psi}_{A^*}(\mathrm{x})\, 1_{\{\mathrm{x} \in A^*\}}$ in the piecewise predictor is replaced by

$$\hat{\psi}_{A_L^*}(\mathrm{x})\, 1_{\{\mathrm{x} \in A_L^*\}} + \hat{\psi}_{A_R^*}(\mathrm{x})\, 1_{\{\mathrm{x} \in A_R^*\}}.$$

If a cell is never split again, its current $\hat{\psi}_A$ is the terminal-node predictor used for prediction.

Remark 2 Although SGT evaluates all cells at each step, model complexity is explicitly controlled. The tree is grown for a user-specified number of splits K (so the final partition has $K + 1$ terminal cells), analogous to pre-pruning by limiting depth or the number of leaves in CART. We also impose standard minimum node-size constraints. Candidate splits are only allowed if both child cells contain at least $m_{\min}$ observations. BSF changes the *order* in which regions are expanded, but for fixed $(K, m_{\min})$ it does not enlarge the hypothesis class beyond trees with at most K splits.

3.1 Class of parametric models for cuts

A key design choice is how candidate cuts are represented. In SGT, cuts are defined by thresholding the output of a parametric model fitted locally within each node. The same nodewise model is also used for within-node prediction, so the fitted model does double duty: it determines the split geometry and it provides the local regression predictor on that cell. Consequently, the model class must balance expressivity with computational tractabil-

ity. It should be rich enough to induce flexible boundaries and accurate prediction, yet fast enough to fit repeatedly during tree growth.

We use models that are linear in parameters, but allow nonlinear dependence on the features through polynomial and interaction terms. Formally, we consider score functions of the form

$$\begin{aligned} \psi_A(\mathrm{x}) = \beta_0 + \sum_{l=1}^{p} \beta_l x_l + \sum_{l=1}^{p} \beta_{ll} x_l^2 + \sum_{1 \le l_1 < l_2 \le p} \beta_{l_1 l_2} x_{l_1} x_{l_2} \\ + \sum_{1 \le l_1 \le l_2 \le l_3 \le p} \beta_{l_1 l_2 l_3} x_{l_1} x_{l_2} x_{l_3} + \cdots, \end{aligned} \tag{2}$$

where the ellipsis indicates additional higher-order terms when included.

In our experiments we typically include terms up to third order, which gives more than enough flexibility while remaining computationally tractable. Coefficients are estimated with the lasso (Tibshirani 1996), encouraging sparse local structure and producing cut boundaries that are less sensitive to noisy (irrelevant) variables.

3.2 Permissible cuts and the space of geometric objects $\mathcal{A}$

A *permissible cut* generalizes the usual tree split. In a classical CART node, left/right assignment is determined by thresholding a single feature. In SGT, left/right assignment is determined by thresholding a fitted model from (2).

Formally, for a node region A, fit a local model to obtain $\hat{\psi}_A$. For any threshold $s \in \mathbb{R}$, define the induced daughters

$$A_L(s) = \{\mathrm{x} \in A : \hat{\psi}_A(\mathrm{x}) \le s\}, \qquad A_R(s) = A \backslash A_L(s).$$

The allowable left daughters form a (node-dependent) class $\mathcal{A}(A) = \{A_L(s) : s \in \mathbb{R}\}$ determined by the chosen model family. To simplify notation we write $\mathcal{A}$ when the dependence on A is clear. We refer to A_L and A_R as the left and right daughters, consistent with standard tree terminology. Figure 2 shows such a cut using an ellipse. The blue elliptical region is a permissible geometric object that becomes the left daughter A_L, i.e. it is the split of A, and the remaining complementary set becomes the right daugter A_R. Another example is given in n Figure 1b which displays rectangular cuts used to split the space.

Elliptical objects are one example of a shape induced by (2). Several familiar split families arise as special cases. The following list summarizes a few.

- *CART splits.* If all coefficients are zero except a single main effect, $\psi_A(\mathrm{x}) = \beta_0 + \beta_l x_l$, then thresholding $\psi_A(\mathrm{x}) \le s$ reduces to an axis-parallel split on x_l at some threshold value (up to swapping the labels of left and right, depending on the sign of β_l).
- *Hyperplane cuts.* If $\psi_A(\mathrm{x}) = \beta_0 + \sum_{l=1}^{p} \beta_l x_l$, then $\psi_A(\mathrm{x}) \le s$ defines a half-space split $\sum_{l=1}^{p} \beta_l x_l \le (s - \beta_0)$.
- *Quadratic (ellipsoidal) cuts.* Using only squared terms, $\psi_A(\mathrm{x}) = \beta_0 + \sum_{l=1}^{p} \beta_{ll} x_l^2$, yields $\sum_{l=1}^{p} \beta_{ll} x_l^2 \le (s - \beta_0)$. When $\beta_{ll} > 0$ for all l and $s - \beta_0 > 0$, this describes an ellipsoid. If some β_{ll} are zero, the set can be unbounded along the corresponding

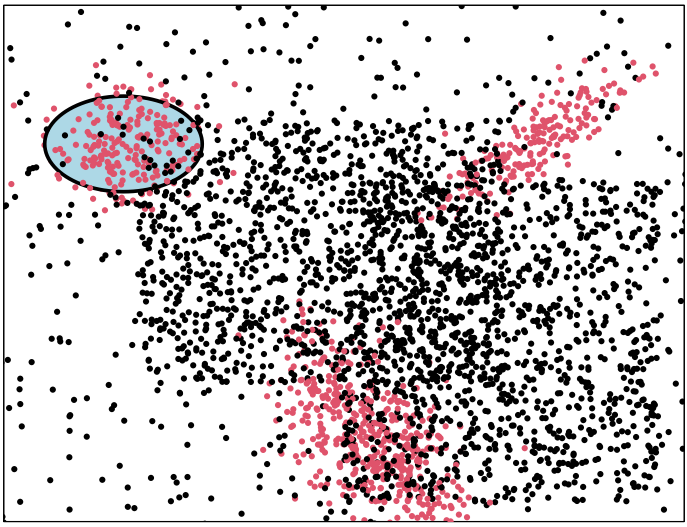


Fig. 2 Elliptical cut within a node region A for two-class data (class membership shown by red and black points). The blue ellipse is a permissible object $A_L \in \mathcal{A}(A)$, producing left daughter A_L and its complement right daughter $A_R = A \setminus A_L$

directions.

- *Oblique quadratic (rotated ellipsoidal) cuts.* More generally, rotated quadratic boundaries can be written as $\mathrm{x}^T\mathrm{Q}\mathrm{x} \le \alpha_0$ with Q symmetric. When Q is positive definite and $\alpha_0 > 0$, this defines an ellipsoid, and cross-terms in (2) correspond to the off-diagonal entries of Q. In coordinates, this corresponds to

$$\psi_A(\mathrm{x}) = \beta_0 + \sum_{l=1}^{p} \beta_{ll} x_l^2 + \sum_{1 \le l_1 < l_2 \le p} \beta_{l_1 l_2} x_{l_1} x_{l_2}.$$

Allowing a nonzero center $(\mathrm{x} - \mu)^T\mathrm{Q}(\mathrm{x} - \mu) \le \alpha_0$ introduces linear terms, yielding

$$\psi_A(\mathrm{x}) = \beta_0 + \sum_{l=1}^{p} \beta_l x_l + \sum_{l=1}^{p} \beta_{ll} x_l^2 + \sum_{1 \le l_1 < l_2 \le p} \beta_{l_1 l_2} x_{l_1} x_{l_2}.$$

More complex cuts involving higher-order polynomial interactions are also possible (Figure 3). Subfigure (a) corresponds to $x_1^2 - x_2^2 - x_3^2 \le \alpha_0$, and subfigure (b) to $x_1 x_2 x_3 \le \alpha_0$. Although we do not pursue it here, one can also induce *interval* cuts of the form $\alpha_1 < \psi_A(\mathrm{x}) \le \alpha_2$. For example, subfigures (c) and (d) use the same score as (b) but apply an interval threshold. Subfigure (e) shows $x_1^2 - x_2^2 - x_3^2 + x_1 x_2 x_3 \le \alpha_0$, and subfigure (f) uses the same form with interval boundaries $\alpha_1 < x_1^2 - x_2^2 - x_3^2 + x_1 x_2 x_3 \le \alpha_2$.

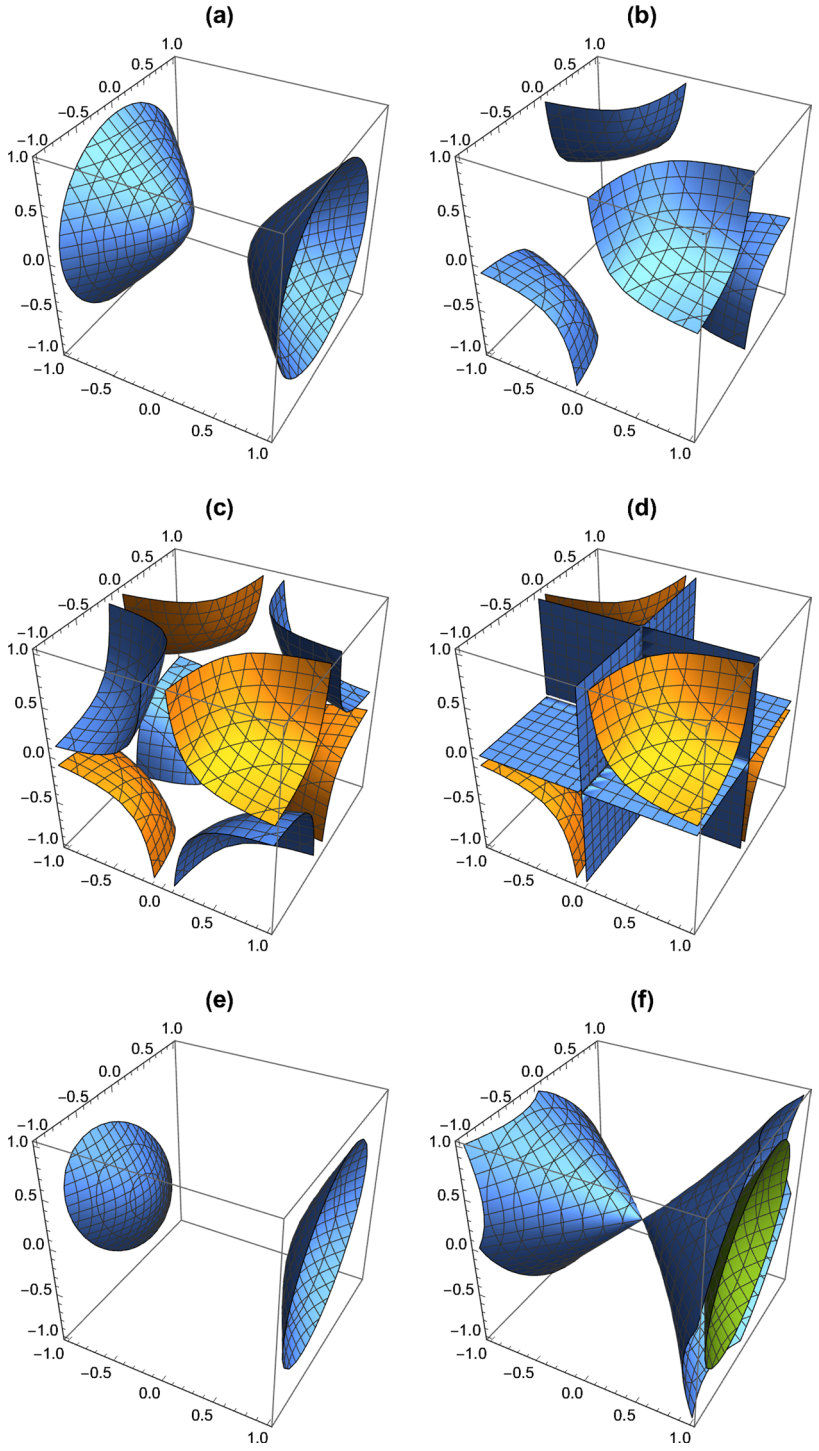


Fig. 3 Examples of geometric cuts induced by the parametric score family (2)

3.3 Splitting rule and empirical risk evaluation

Within a node A, the fitted model $\hat{\psi}_A$ from the lasso induces a one-dimensional score $z = \hat{\psi}_A(\mathrm{x})$, and candidate cuts correspond to thresholding this score. This groups observations with similar lasso-predicted values and yields partitions that reflect the structure of the fitted model in that node. The lasso also controls local complexity by selecting a sparse subset of terms in (2). This leads to simpler boundaries in small nodes, with more expressive boundaries appearing when the data support them.

Let $I(A) = \{i : \mathrm{x}_i \in A\}$ and $M = M(A) = |I(A)|$. For convenience, re-index the observations in A as $\{(\mathrm{x}_{A,m}, y_{A,m})\}_{m=1}^{M}$ and define the fitted scores $\hat{y}_{A,m} = \hat{\psi}_A(\mathrm{x}_{A,m})$. Order the scores:

$$\hat{y}_A^{(1)} \le \hat{y}_A^{(2)} \le \cdots \le \hat{y}_A^{(M)}.$$

Given $\hat{\psi}_A$, a natural threshold search considers the $M-1$ cutpoints between consecutive ordered scores. Evaluating the full risk reduction for each cutpoint would require repeated refits within the candidate daughters, which is computationally expensive. We therefore select a threshold using a fast one-dimensional criterion computed from the ordered scores, and then evaluate empirical risk reduction using refitted daughter models.

For $m \in \{1, \ldots, M-1\}$, define the split statistic

$$\hat{\delta}_A(m) = \sum_{i \le m} \left(\hat{y}_A^{(i)} - \frac{1}{m} \sum_{j \le m} \hat{y}_A^{(j)} \right)^2 + \sum_{i > m} \left(\hat{y}_A^{(i)} - \frac{1}{M-m} \sum_{j > m} \hat{y}_A^{(j)} \right)^2. \tag{3}$$

This is the within-group sum of squares obtained by splitting the ordered scores into $\{\hat{y}_A^{(1)}, \ldots, \hat{y}_A^{(m)}\}$ and $\{\hat{y}_A^{(m+1)}, \ldots, \hat{y}_A^{(M)}\}$ and using a constant fit within each group. We then choose

$$m^* \in \arg \min_{m_{\min} \le m \le M - m_{\min}} \hat{\delta}_A(m), \qquad s = \frac{1}{2}\left(\hat{y}_A^{(m^*)} + \hat{y}_A^{(m^*+1)} \right),$$

and define the induced daughters

$$A_L = \{\mathrm{x} \in A : \hat{\psi}_A(\mathrm{x}) \le s\}, \qquad A_R = \{\mathrm{x} \in A : \hat{\psi}_A(\mathrm{x}) > s\}.$$

To evaluate the empirical risk reduction from splitting A, we refit the lasso separately within each daughter, yielding $\hat{\psi}_{A_L}$ and $\hat{\psi}_{A_R}$. The (squared-error) empirical risk reduction is

$$R(A) = \sum_{\mathrm{x}_i \in A} (y_i - \hat{\psi}_A(\mathrm{x}_i))^2 - \left\{ \sum_{\mathrm{x}_i \in A_L} (y_i - \hat{\psi}_{A_L}(\mathrm{x}_i))^2 + \sum_{\mathrm{x}_i \in A_R} (y_i - \hat{\psi}_{A_R}(\mathrm{x}_i))^2 \right\}.$$

At step k, $R(A)$ is computed for all cells $A \in \Pi_{k-1}$, and the procedure selects the cell A^* that maximizes $R(A)$. The corresponding daughters replace A^* to form Π_k, and the tree predictor is updated by replacing the term $\hat{\psi}_{A^*}(\mathrm{x})\, 1_{\{\mathrm{x} \in A^*\}}$ with $\hat{\psi}_{A_L^*}(\mathrm{x})\, 1_{\{\mathrm{x} \in A_L^*\}} + \hat{\psi}_{A_R^*}(\mathrm{x})\, 1_{\{\mathrm{x} \in A_R^*\}}$. The procedure repeats until K splits have been performed.

3.4 Partitioning number and the size of the tree class

SGTs permit multivariate geometric cuts, so the induced class of tree partitions can be far richer than that of CART. To quantify the size of this partition space, we use two standard combinatorial quantities from statistical learning theory. The first is the *shatter coefficient* of the underlying cut class, and the second is the *partitioning number* of the resulting *k*-split tree class.

Definition 1 Let $\zeta_1, \ldots, \zeta_n$ be vectors in $\mathbb{R}^p$, and let $\mathcal{A}$ be a collection of sets in $\mathbb{R}^p$. Define

$$N(\mathcal{A}, \zeta_1, \ldots, \zeta_n) := \Big| \big\{ \{\zeta_1, \ldots, \zeta_n\} \cap A : \ A \in \mathcal{A} \big\} \Big|,$$

the number of distinct subsets of $\{\zeta_1, \ldots, \zeta_n\}$ induced by intersections with $\mathcal{A}$. The *n*-shatter coefficient of $\mathcal{A}$ is

$$s(\mathcal{A}, n) = \max_{\zeta_1, \ldots, \zeta_n \in \mathbb{R}^p} N(\mathcal{A}, \zeta_1, \ldots, \zeta_n).$$

For convenience, set $s(\mathcal{A}, 0) := 0$.
The quantity $s(\mathcal{A}, n)$ is the maximum number of different subsets of *n* points in $\mathbb{R}^p$ that can be realized as $S \cap A$ with $A \in \mathcal{A}$. It is useful to keep in mind the following examples.

E1. Let $\mathcal{A}$ be the class of all axis-parallel *p*-dimensional rectangles. Then $s(\mathcal{A}, n) \leq n^{2p+1}$ for $n > 4p$ Devroye et al.(1996, Theorems 13.3 and 13.8).
E2. Let $\mathcal{A}$ be the class of all half-spaces in $\mathbb{R}^p$. Then $s(\mathcal{A}, n) \leq n^p$ (Cover 1965).
E3. By Devroye et al.(1996, Theorem 13.9) (see also Cover (1965); Steele (1975); Dudley (1978)), if $\mathcal{V}$ is a finite-dimensional real vector space of functions on $\mathbb{R}^p$ with $\dim(\mathcal{V}) = V \geq 1$, then the class $\{ \{\mathrm{x} : g(\mathrm{x}) \leq 0\} : g \in \mathcal{V} \}$ has shatter coefficient at most $n^V + 1 \leq n^{V+1}$.

These examples show how the shatter coefficient measures the richness of a *single* cut family. For trees, we also need a notion that counts how many distinct partitions can be obtained by applying such cuts successively.

Let $\mathcal{P}_k = \mathcal{P}_k(\mathcal{A})$ denote the family of partitions of $\mathbb{R}^p$ into $k+1$ cells obtainable by starting from $\Pi_0 = \{\mathbb{R}^p\}$ and performing *k* successive binary splits, each split replacing one current cell A by $A \cap T$ and $A \cap T^c$ for some $T \in \mathcal{A}$. For design points $\mathrm{x}_1, \ldots, \mathrm{x}_n \in \mathbb{R}^p$, write $S = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ and define the restriction of a partition Π to S by

$$\Pi \cap S := \{A \cap S : \ A \in \Pi, \ A \cap S \neq \emptyset\}.$$

Let $\rho_x(\mathcal{P}_k, S)$ be the number of distinct restricted partitions $\Pi \cap S$ over $\Pi \in \mathcal{P}_k$. The *partitioning number* (Lugosi and Nobel 1996) is

$$\rho(\mathcal{P}_k, n) = \max_{\mathrm{x}_1, \ldots, \mathrm{x}_n \in \mathbb{R}^p} \rho_x(\mathcal{P}_k, \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}),$$

the maximum number of distinct clusterings of n points induced by k-split partitions from $\mathcal{P}_k$.

Theorem 1 *If the shatter coefficient $s(\mathcal{A}, n)$ is convex in n, then $\rho(\mathcal{P}_k, n) \leq s(\mathcal{A}, n)^k$.*

Theorem 1 shows that, after k splits, an upper bound on the combinatorial size of the induced tree class is given by the shatter coefficient of the base cut family. Although $s(\mathcal{A}, n)$ is only an upper bound, it makes clear how a richer geometric cut class can enlarge the collection of candidate partitions available to a greedy risk-reduction procedure. For intuition, consider rectangles in $\mathbb{R}^2$ with $n = 4$ points. In this case, rectangles can realize all $2^4 = 16$ subsets of the four points, so $s(\mathcal{A}, 4) = 16$ (Figure 4).

In terms of growth rates, half-spaces in $\mathbb{R}^p$ satisfy $s(\mathcal{A}, n) = O(n^p)$ (Example E2). Coordinate-threshold (CART) stumps are contained in this family, so their shatter coefficient is no larger. Axis-parallel rectangles satisfy $s(\mathcal{A}, n) \leq n^{2p+1}$ (Example E1). Finally, if $\mathcal{A}$ is induced by a finite-dimensional parametric score family such as (2) truncated at degree three, then Example E3 yields $s(\mathcal{A}, n) \leq n^{V+1}$ with $V = O(p^3)$. Therefore $\rho(\mathcal{P}_k, n) \leq n^{O(kp^3)}$. This growth helps explain why permitting richer cut geometries can allow larger empirical risk reductions than coordinate-threshold splits and purely linear cut families.

Proof Fix design points $\mathrm{x}_1, \ldots, \mathrm{x}_n$ and write $S = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$. For $k = 1$, each partition in $\mathcal{P}_1$ corresponds to choosing $T \in \mathcal{A}$ and splitting $\mathbb{R}^p$ into T and T^c. The number of distinct restricted two-cell partitions of S is therefore at most $N(\mathcal{A}, \mathrm{x}_1, \ldots, \mathrm{x}_n) \leq s(\mathcal{A}, n)$.

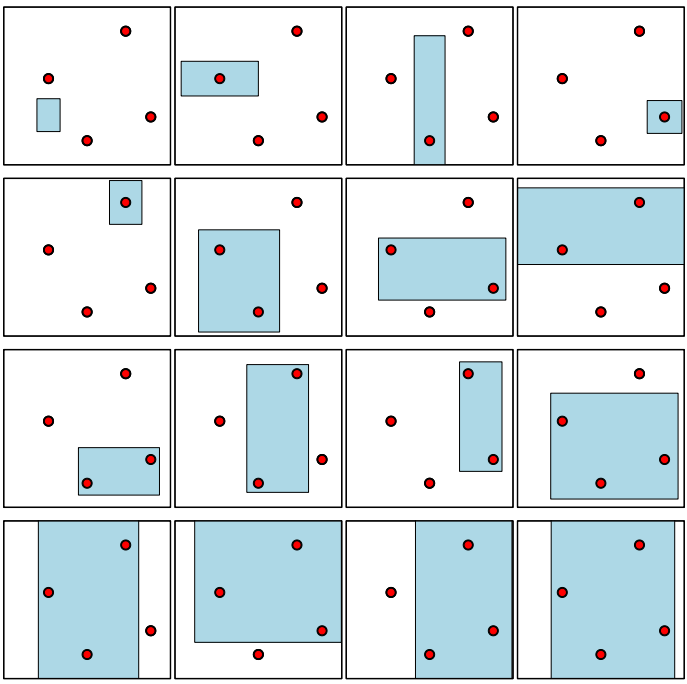


Fig. 4 $2^4 = 16$ rectangles separating $n = 4$ points in $p = 2$ dimensions

Now suppose $k \geq 2$ and consider any restricted partition of S arising from $\mathcal{P}_{k-1}$. Let its nonempty cells be $A_1, \ldots, A_r$, with $r \leq k$ and with $m_1, \ldots, m_r$ points in the cells ($\sum_{j=1}^{r} m_j = n$). To refine this partition by one additional split, we choose one nonempty cell A_j and split it using some $T \in \mathcal{A}$. The number of distinct refinements obtainable by splitting A_j is at most $s(\mathcal{A}, m_j)$, so the total number of distinct refinements of the restricted partition is bounded by $\sum_{j=1}^{r} s(\mathcal{A}, m_j)$.

Convexity of $s(\mathcal{A}, \cdot)$ implies superadditivity (Bruckner 1962). For $0 \leq m \leq n$, letting $\lambda = m/n$, convexity and $s(\mathcal{A}, 0) = 0$ give $s(\mathcal{A}, m) = s(\mathcal{A}, \lambda n) \leq \lambda s(\mathcal{A}, n)$ and $s(\mathcal{A}, n-m) \leq (1-\lambda) s(\mathcal{A}, n)$, hence $s(\mathcal{A}, m) + s(\mathcal{A}, n-m) \leq s(\mathcal{A}, n)$. Iterating this inequality yields $\sum_{j=1}^{r} s(\mathcal{A}, m_j) \leq s(\mathcal{A}, n)$. Therefore each restricted $(k-1)$-split partition of S admits at most $s(\mathcal{A}, n)$ distinct restricted k-split refinements, and so $\rho_x(\mathcal{P}_k, S) \leq s(\mathcal{A}, n)\, \rho_x(\mathcal{P}_{k-1}, S)$. Iterating from the base case $k = 1$ gives $\rho_x(\mathcal{P}_k, S) \leq s(\mathcal{A}, n)^k$, and maximizing over S yields the claim. □

3.5 Greediness of splits and empirical risk minimization

Theorem 1 bounds the number of distinct k-split partitions that can be induced on n points. Together with the examples in the previous subsection, it indicates that permitting multivariate cuts can greatly increase the number of data partitions available after k splits. This added flexibility can lower approximation error, but it can also increase estimation error. We first make the approximation gain explicit for a single split, and then relate empirical and population risk through partitioning numbers and within-leaf model complexity. This shows that, when tree size and within-leaf model complexity are controlled, the added flexibility of SGTs can translate into smaller population risk and improved test performance.

3.5.1 One-split families

Consider two one-split regression classes.

- $\mathcal{H}^{(1)}_{\mathrm{CART}}$ is the class of all one-split CART regressors with a coordinate-threshold cut and constant predictions in each daughter,

$$f(\mathrm{x}) = c_L\, 1_{\{x_l \leq s\}} + c_R\, 1_{\{x_l > s\}}, \qquad l \in \{1, \ldots, p\},\ s \in \mathbb{R},\ c_L, c_R \in \mathbb{R}.$$

- $\mathcal{H}^{(1)}_{\mathrm{SGT}}$ is the class of all one-split SGT regressors obtained by choosing a permissible cut $A_L \in \mathcal{A}$ of the form $A_L = \{\mathrm{x} : g(\mathrm{x}) \leq s\}$ for some g from the parametric score class (2) and threshold $s \in \mathbb{R}$, with $A_R = \mathcal{X} \setminus A_L$, together with within-region predictors φ_L, φ_R from the same parametric family,

$$f(\mathrm{x}) = \varphi_L(\mathrm{x})\, 1_{\{\mathrm{x} \in A_L\}} + \varphi_R(\mathrm{x})\, 1_{\{\mathrm{x} \in A_R\}}.$$

We compare the best achievable population risk in these two classes. For a function class $\mathcal{H}$, we refer to $\inf_{f \in \mathcal{H}} R(f)$ as its oracle risk, the smallest population risk attainable within $\mathcal{H}$.

Proposition 1 *Let $R(f) = \mathbb{E}[(Y - f(\mathrm{X}))^2]$ denote population risk under squared loss, where $Y = \psi(\mathrm{X}) + \varepsilon$, $\psi(\mathrm{x}) = \mathbb{E}[Y \mid \mathrm{X} = \mathrm{x}]$, $\mathbb{E}[\varepsilon \mid \mathrm{X}] = 0$, and $\mathbb{E}[\varepsilon^2 \mid \mathrm{X}] = \sigma^2$. Assume:*

(i) *The parametric class in (2) contains all constants and all single-coordinate linear*

functions x_l, so $\mathcal{H}^{(1)}_{\rm CART} \subseteq \mathcal{H}^{(1)}_{\rm SGT}$.

(ii) *The true regression function ψ belongs to this parametric class.*

Then

$$\inf_{f\in\mathcal{H}^{(1)}_{\rm CART}} R(f) - \inf_{f\in\mathcal{H}^{(1)}_{\rm SGT}} R(f) = \inf_{f\in\mathcal{H}^{(1)}_{\rm CART}} \mathbb{E}[(\psi(\mathrm{X}) - f(\mathrm{X}))^2] \geq 0,$$

with equality if and only if ψ coincides almost surely with an element of $\mathcal{H}^{(1)}_{\rm CART}$.

Proof Under squared loss,

$$R(f) = \mathbb{E}[(\psi(\mathrm{X}) - f(\mathrm{X}))^2] + \sigma^2.$$

By (ii), $\psi \in \mathcal{H}^{(1)}_{\rm SGT}$ (take $\varphi_L = \varphi_R = \psi$), hence $\inf_{f\in\mathcal{H}^{(1)}_{\rm SGT}} R(f) = \sigma^2$. The identity follows immediately, and the equality condition is exactly $\inf_{f\in\mathcal{H}^{(1)}_{\rm CART}} \mathbb{E}[(\psi(\mathrm{X}) - f(\mathrm{X}))^2] = 0$. □

Proposition 1 shows that enlarging the cut family and the within-region model family cannot worsen oracle risk. The improvement is exactly the L_2 approximation error of the CART one-split class,

$$\mathcal{E}_{\rm CART}(\psi) = \inf_{f\in\mathcal{H}^{(1)}_{\rm CART}} \mathbb{E}[(\psi(\mathrm{X}) - f(\mathrm{X}))^2].$$

Whenever ψ cannot be represented by a single coordinate-threshold split with two constants, $\mathcal{E}_{\rm CART}(\psi) > 0$ and the oracle SGT one-split class achieves strictly smaller population risk.

To connect this with empirical risk, define

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathrm{x}_i))^2.$$

Writing $y_i = \psi(\mathrm{x}_i) + \varepsilon_i$ and taking expectation conditional on $\mathrm{x}_1, \ldots, \mathrm{x}_n$ yields, for each fixed f,

$$\mathbb{E}[R_n(f) \mid \mathrm{x}_1, \ldots, \mathrm{x}_n] = \frac{1}{n}\sum_{i=1}^{n}(\psi(\mathrm{x}_i) - f(\mathrm{x}_i))^2 + \sigma^2.$$

Thus the conditional expected empirical risk depends on f only through its squared error on the observed design points. This motivates the empirical approximation errors

$$\mathcal{E}_{{\rm CART},n}(\psi) = \inf_{f\in\mathcal{H}^{(1)}_{\rm CART}} \frac{1}{n}\sum_{i=1}^{n}(\psi(\mathrm{x}_i) - f(\mathrm{x}_i))^2, \qquad \mathcal{E}_{{\rm SGT},n}(\psi) = \inf_{f\in\mathcal{H}^{(1)}_{\rm SGT}} \frac{1}{n}\sum_{i=1}^{n}(\psi(\mathrm{x}_i) - f(\mathrm{x}_i))^2.$$

Since $\mathcal{H}^{(1)}_{\rm CART} \subseteq \mathcal{H}^{(1)}_{\rm SGT}$, we always have $\mathcal{E}_{{\rm SGT},n}(\psi) \leq \mathcal{E}_{{\rm CART},n}(\psi)$.

Actual fitting minimizes $R_n(f)$ using the observed responses, so a richer class can also reduce $R_n(f)$ by fitting noise. To ensure that empirical risk reductions translate into population improvements, we control the uniform deviation

$$\sup_{f \in \mathcal{H}} |R_n(f) - R(f)|.$$

Indeed, for any class $\mathcal{H}$,

$$\left| \inf_{f \in \mathcal{H}} R_n(f) - \inf_{f \in \mathcal{H}} R(f) \right| \le \sup_{f \in \mathcal{H}} |R_n(f) - R(f)|,$$

so uniform control of $|R_n - R|$ implies that the empirical and population oracle risks are close.

3.5.2 Generalization control via partitioning numbers

A richer split family can lower approximation error because it enlarges the set of partitions the tree can express. The same enlargement can also increase estimation error because there are more ways to fit a finite sample. To ensure that decreases in training risk correspond to improved population performance, we control the uniform gap between empirical and population risk over the class of *k*-split trees.

For simplicity, assume a bounded problem with $|Y| \le B$, and suppose all candidate predictors are clipped so $|f(\mathrm{x})| \le B$ for every $f \in \mathcal{H}^{(k)}$ and $\mathrm{x} \in \mathcal{X}$. Let $\mathcal{P}_k$ denote the family of partitions induced by *k* splits (with at most $k+1$ cells), and let $\mathcal{H}^{(k)}$ denote the associated class of *k*-split trees. Terminal node predictors are drawn from a fixed bounded within-node family $\mathcal{F}$ (for example, clipped lasso fits). Write $d_{\mathrm{leaf}} = d_{\mathrm{leaf}}(\mathcal{F})$ for a finite complexity index of $\mathcal{F}$, such as its pseudo-dimension (VC-subgraph dimension). Under the fixed-dictionary setting considered here (fixed `hcut` and fixed feature dimension *p*), d_{leaf} is a constant determined by the chosen within-node model class and does not depend on *n*.

For partition-based classes, a VC-type deviation inequality (see, for example, Lugosi and Nobel(1996, Proposition 1 and Lemma 1) and Lugosi and Nobel (1996, Lemma 1)) gives a tail bound of the form

$$\mathbb{P}\left(\sup_{f \in \mathcal{H}^{(k)}} |R_n(f) - R(f)| > t \right) \le C_0 \, \rho(\mathcal{P}_k, n) \, \phi(t)^{k+1} \exp\left(-c_0 \frac{nt^2}{B^4} \right),$$

where $\phi(t)$ captures the complexity of the within-node family $\mathcal{F}$ through a covering number (or entropy) bound. For many bounded parametric families, $\phi(t)$ can be taken to be polynomial in $1/t$, with degree controlled by d_{leaf}. In our setting, $\mathcal{F}$ is a class of regularized linear predictors over a fixed dictionary of basis terms, so one can appeal to empirical covering-number results; see, for example, Zhang (2002).

Inverting the tail bound yields that for any $\delta \in (0, 1)$,

$$\sup_{f\in\mathcal{H}^{(k)}} |R_n(f) - R(f)| \;\lesssim\; B^2\sqrt{\frac{\log\rho(\mathcal{P}_k,n) \;+\; (k+1)\,d_{\text{leaf}}\log n \;+\; \log(1/\delta)}{n}} \tag{4}$$

with probability at least $1-\delta$. In particular, for any (data-dependent) estimator $\hat{f}\in\mathcal{H}^{(k)}$,

$$R(\hat{f}) \le R_n(\hat{f}) + C\,B^2\sqrt{\frac{\log\rho(\mathcal{P}_k,n) \;+\; (k+1)\,d_{\text{leaf}}\log n \;+\; \log(1/\delta)}{n}},$$

and if $\hat{f}^{\text{ERM}} \in \arg\min_{f\in\mathcal{H}^{(k)}} R_n(f)$, then

$$R(\hat{f}^{\text{ERM}}) \le \inf_{f\in\mathcal{H}^{(k)}} R(f) + 2C\,B^2\sqrt{\frac{\log\rho(\mathcal{P}_k,n) \;+\; (k+1)\,d_{\text{leaf}}\log n \;+\; \log(1/\delta)}{n}}.$$

By Theorem 1, $\log\rho(\mathcal{P}_k,n) \le k\log s(\mathcal{A},n)$. Hence, if $s(\mathcal{A},n)$ grows polynomially in n (as in Examples E1–E3), then $\log\rho(\mathcal{P}_k,n) = O(k\log n)$ and the estimation term in (4) scales as

$$O\left(\sqrt{\frac{(k+(k+1)\,d_{\text{leaf}})\log n}{n}}\right).$$

In particular, if $k=k_n$ satisfies $k_n\log n/n\to 0$, then empirical and population risks are uniformly close over $\mathcal{H}^{(k_n)}$. Under this type of scaling, richer cut families can reduce approximation error without sacrificing generalization, provided tree size is controlled. Equivalently, if increasing `hcut` lowers the oracle term $\inf_{f\in\mathcal{H}^{(k_n)}} R(f)$, then the population risk of the empirically fit tree can also decrease.

3.5.3 A simple example with an oblique split

To give an explicit illustration of how SGTs can reduce risk, consider $p=2$ with $\mathrm{X}=(X_1,X_2)$ uniformly distributed on $[-1,1]^2$ and

$$\psi(\mathrm{x}) = \begin{cases} +1, & x_1+x_2>0,\\ -1, & x_1+x_2\le 0. \end{cases}$$

This is a two-region model separated by the oblique hyperplane $x_1+x_2=0$. A one-split SGT with permissible cuts including $\{x_1+x_2\le 0\}$ represents ψ exactly by assigning constants -1 and $+1$ in the two daughters. Hence

$$\mathcal{E}_{\text{SGT}}(\psi) := \inf_{f\in\mathcal{H}^{(1)}_{\text{SGT}}} \mathbb{E}[(\psi(\mathrm{X})-f(\mathrm{X}))^2] = 0, \qquad \inf_{f\in\mathcal{H}^{(1)}_{\text{SGT}}} R(f) = \sigma^2.$$

We now compute the best one-split CART approximation. By symmetry it suffices to consider coordinate-threshold splits of the form $\{X_1\le s\}$ with $s\in(-1,1)$. Let $L=\{X_1\le s\}$ and $R=\{X_1>s\}$, with optimal constants $c_L(s)=\mathbb{E}[\psi(\mathrm{X})\mid L]$ and $c_R(s)=\mathbb{E}[\psi(\mathrm{X})\mid R]$. For $x_1\in[-1,1]$,

$$\mathbb{P}(\psi(\mathrm{X}) = +1 \mid X_1 = x_1) = \mathbb{P}(X_2 > -x_1) = \frac{1+x_1}{2}, \qquad \mathbb{P}(\psi(\mathrm{X}) = -1 \mid X_1 = x_1) = \frac{1-x_1}{2},$$

so

$$\mathbb{E}[\psi(\mathrm{X}) \mid X_1 = x_1] = \frac{1+x_1}{2} - \frac{1-x_1}{2} = x_1.$$

Therefore,

$$\begin{aligned} c_L(s) =& \mathbb{E}[\psi(\mathrm{X}) \mid X_1 \le s] = \mathbb{E}[X_1 \mid X_1 \le s] = \frac{s-1}{2} \\ c_R(s) =& \mathbb{E}[\psi(\mathrm{X}) \mid X_1 > s] = \mathbb{E}[X_1 \mid X_1 > s] = \frac{s+1}{2}. \end{aligned}$$

Because $\psi(\mathrm{X}) \in \{-1, 1\}$, we have $\mathrm{Var}(\psi(\mathrm{X}) \mid \cdot) = 1 - \mathbb{E}[\psi(\mathrm{X}) \mid \cdot]^2$, so

$$\mathrm{Var}(\psi(\mathrm{X}) \mid X_1 \le s) = 1 - \left(\frac{s-1}{2}\right)^2, \qquad \mathrm{Var}(\psi(\mathrm{X}) \mid X_1 > s) = 1 - \left(\frac{s+1}{2}\right)^2.$$

Using $P(X_1 \le s) = (s+1)/2$ and $P(X_1 > s) = (1-s)/2$, the approximation error of a one-split CART tree at threshold s is

$$\begin{aligned} \mathcal{E}_{\mathrm{CART}}(\psi; s) &= \mathbb{E}[(\psi(\mathrm{X}) - c_L(s))^2 1_{\{X_1 \le s\}}] + \mathbb{E}[(\psi(\mathrm{X}) - c_R(s))^2 1_{\{X_1 > s\}}] \\ &= \frac{s+1}{2} \mathrm{Var}(\psi(\mathrm{X}) \mid X_1 \le s) + \frac{1-s}{2} \mathrm{Var}(\psi(\mathrm{X}) \mid X_1 > s) \\ &= \frac{s^2}{4} + \frac{3}{4}. \end{aligned}$$

This is minimized at $s = 0$, giving $\mathcal{E}_{\mathrm{CART}}(\psi) = 3/4$. The same value arises for splits on X_2, so

$$\inf_{f \in \mathcal{H}^{(1)}_{\mathrm{CART}}} R(f) = \sigma^2 + \frac{3}{4}.$$

Therefore the oracle SGT one-split tree achieves risk σ^2, whereas the oracle CART one-split tree must incur an additional $\mathcal{E}_{\mathrm{CART}}(\psi) = 3/4$ units of squared error because its partitions cannot represent the oblique boundary. For example, if $\sigma^2 = 1/4$, then the best one-split CART tree has risk 1, while the oracle SGT tree achieves risk 1/4.

4 Empirical results

The previous section shows that enlarging the class of permissible cuts can reduce approximation error and, with appropriate complexity control, improve prediction performance. In practice, however, the best-split-first (BSF) search used by SGTs evaluates many candidate splits and can overfit if the tree is grown too large or if the local split models are unstable.

Our implementation therefore includes several stabilization steps. We cap the number of splits at K, we enforce minimum node sizes $m_{\min}$, we use lasso regularization for all local models, we apply feature filtering to reduce the candidate dictionary, we select the cut-family index `hcut` (defined below) by cross-validation, and we include an out-of-bag (OOB) stability guard that falls back to CART when a model-based split fails to improve held-out error. Algorithm 2 summarizes the full procedure.

To separate the effect of split geometry from the stabilization components, the experiments in this section are organized into two groups. Sections 4.3–4.5 provide controlled illustrations designed to compare directly to CART. In these illustrations, the cut family is fixed in advance and the adaptive components (automatic `hcut` selection and the OOB stability guard) are disabled. Sections 4.6–4.7 report large-scale synthetic and real-data benchmarking studies using our recommended pipeline, which includes filtering, adaptive `hcut` selection, and the OOB-based stability guard.

4.1 Software and model families

4.1.1 Software

All experiments were conducted using the R package `randomForestSGT`, developed for implementing SGTs. Local lasso models are fit by coordinate descent, with the penalty parameter chosen by 10-fold cross-validation. Tree growth is controlled by the number of splits K, and all candidate splits must satisfy the minimum node-size constraints $M(A_L) \geq m_{\min}$ and $M(A_R) \geq m_{\min}$ (as in Algorithm 1), where $M(A)$ denotes the sample size of cell A.

4.1.2 Model families indexed by `hcut`

Candidate split functions are generated from the parametric expansion in (2). In our implementation, `hcut` indexes a nested sequence of polynomial families with increasing richness:

0. `hcut` $= 0$: CART axis-aligned splits.
1. `hcut` $= 1$: intercept plus linear terms $\{x_l\}_{l=1}^{p}$ (hyperplane cuts).
2. `hcut` $= 2$: adds quadratic main effects $\{x_l^2\}_{l=1}^{p}$.
3. `hcut` $= 3$: adds all pairwise interactions $\{x_{l_1}x_{l_2}\}_{l_1<l_2}$ (full quadratic forms).
4. `hcut` $= 4$: adds cubic monomials involving at most two variables (e.g., x_l^3, $x_l^2 x_m$, $x_l x_m^2$).
5. `hcut` $= 5$: adds selected quartic monomials involving up to three variables (e.g., $x_l^2 x_m x_r$).
6. `hcut` $= 6$: adds all three-way interactions $\{x_{l_1}x_{l_2}x_{l_3}\}_{l_1<l_2<l_3}$.

Each level contains all terms from earlier levels. Increasing `hcut` enlarges the candidate dictionary and can reduce bias, but it can increase variance in small nodes and under strong collinearity. This motivates the safeguards described next.

Require: Data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$; number of splits K; minimum node size $m_{\min}$; cut index `hcut`; optional in-bag/OOB split; optional OOB guard.

(Optional) In-bag/OOB split

Choose in-bag indices $\mathcal{I}$ (for example, a bootstrap sample) and set $\mathcal{O} = \{1, \ldots, n\} \setminus \mathcal{I}$. (If not used, set $\mathcal{I} = \{1, \ldots, n\}$ and $\mathcal{O} = \emptyset$.)

Filtering

if p is large **then**
  Grow a shallow pilot SGT with `hcut` $= 1$ on $\mathcal{I}$.
  Keep predictors with a nonzero coefficient in at least one terminal-cell lasso fit.
end if
Grow a second shallow pilot SGT with target `hcut` on $\mathcal{I}$ and filter again.

Best-split-first (BSF) induction

Initialize partition $\Pi_0 = \{\mathcal{X}\}$.
For each cell A, initialize a cut-type flag $\tau(A) = \text{hcut}$.
for $k = 1$ to K **do**
  for each terminal cell $A \in \Pi_{k-1}$ **do**
    Let $M_{\mathcal{I}}(A) = |I \cap A|$.
    if $M_{\mathcal{I}}(A) < 2m_{\min}$ **then**
      Mark A as not splittable and continue.
    end if
    if $\tau(A) > 0$ **then**
      Fit lasso model $\hat{\psi}_A$ on $\mathcal{I} \cap A$ using family $\tau(A)$.
      Compute fitted values $\hat{y}_i = \hat{\psi}_A(\mathbf{x}_i)$ for $i \in \mathcal{I} \cap A$.
      Choose threshold index m^* minimizing (3) over $m \in \{m_{\min}, \ldots, M_{\mathcal{I}}(A) - m_{\min}\}$ and form daughters (A_L, A_R).
      Fit lasso models $\hat{\psi}_{A_L}$ and $\hat{\psi}_{A_R}$ on $\mathcal{I} \cap A_L$ and $\mathcal{I} \cap A_R$.
      Compute in-bag risk reduction $R_{\mathcal{I}}(A)$ for this candidate split.
    else
      Compute the best CART coordinate-threshold split at A using $\mathcal{I} \cap A$.
      Fit CART daughter predictors on $\mathcal{I} \cap A_L$ and $\mathcal{I} \cap A_R$ using sample averages.
      Compute in-bag risk reduction $R_{\mathcal{I}}(A)$ for this CART candidate split.
    end if
    if OOB guard is enabled and $\mathcal{O}(A) \neq \emptyset$ **then**
      Compute $R_{\text{OOB}}^{\text{SGT}}(A)$ for the current candidate split.
      Compute $R_{\text{OOB}}^{\text{CART}}(A)$ for the best CART coordinate-threshold split at A.
      if $R_{\text{OOB}}^{\text{CART}}(A) > R_{\text{OOB}}^{\text{SGT}}(A)$ **then**
        Replace the candidate split at A by the CART split.
        Set $\tau(A) = 0$ and enforce $\tau(\cdot) = 0$ for all future descendants of A.
      end if
    end if
  end for
  Choose $A^* = \arg\max_{A \in \Pi_{k-1}} R_{\mathcal{I}}(A)$ and split it into (A_L^*, A_R^*).
  Set $\tau(A_L^*) = \tau(A^*)$ and $\tau(A_R^*) = \tau(A^*)$.
  Update $\Pi_k = (\Pi_{k-1} \setminus \{A^*\}) \cup \{A_L^*, A_R^*\}$.
end for

Output

return final partition Π_K and fitted leaf models $\{\hat{\psi}_A\}_{A \in \Pi_K}$.

Algorithm 2 SGT implementation with filtering and OOB stability guard

4.2 Filtering, stability safeguards, and complexity control

4.2.1 Feature filtering

To improve runtime and reduce variance when p is large, we apply a filtering step to reduce the effective predictor set before fitting the final tree. Each terminal cell A yields a lasso fit with coefficient vector $\hat{\beta}_A$. In high-dimensional settings, we first grow a shallow pilot tree with $\texttt{hcut} = 1$ and retain predictors that appear with a nonzero coefficient in at least one terminal-cell fit. We then grow a second shallow pilot tree using the target $\texttt{hcut}$ and filter again in the same way. This reduces the effective dimension from p to p_F used in (2), shrinking the candidate dictionary and improving stability. In low dimensions, the first pilot step is omitted.

4.2.2 Choosing hcut

The parameter $\texttt{hcut}$ controls a bias–variance tradeoff. Larger $\texttt{hcut}$ can reduce bias by enabling more flexible boundaries, but it increases the candidate dictionary and can increase variance in small nodes. In the benchmarking studies (Sections 4.6–4.7), our default strategy selects $\texttt{hcut}$ by cross-validation over a prespecified candidate set. In the controlled illustrations (Sections 4.3–4.5), $\texttt{hcut}$ is fixed in advance to isolate the effect of cut geometry.

4.2.3 OOB stability guard against split-selection bias

BSF search evaluates many candidate splits across all current cells and selects the globally best one. Even when K is fixed, this global maximization can introduce optimistic bias in training-set risk reduction, especially when local lasso fits are unstable, for example in small cells or under strong collinearity. We mitigate this effect using an OOB stability guard whenever OOB data are available, such as when growing a tree on a bootstrap in-bag sample. This guard is enabled in the benchmarking studies (Sections 4.6–4.7).

Consider a tree grown on in-bag indices $\mathcal{I} \subset \{1, \ldots, n\}$, with OOB indices $\mathcal{O} = \{1, \ldots, n\} \setminus \mathcal{I}$. For a cell A, define $\mathcal{O}(A) = \{i \in \mathcal{O} : \mathrm{x}_i \in A\}$, and similarly for $\mathcal{O}(A_L)$ and $\mathcal{O}(A_R)$. Let (A_L, A_R) be a candidate split learned from the in-bag data in A, with fitted models $\hat{\psi}_A$, $\hat{\psi}_{A_L}$, and $\hat{\psi}_{A_R}$ trained on $\mathcal{I} \cap A$, $\mathcal{I} \cap A_L$, and $\mathcal{I} \cap A_R$, respectively. The OOB risk reduction of this split is

$$R_{\mathrm{OOB}}(A) = \sum_{i \in \mathcal{O}(A)} (y_i - \hat{\psi}_A(\mathrm{x}_i))^2 - \left[\sum_{i \in \mathcal{O}(A_L)} (y_i - \hat{\psi}_{A_L}(\mathrm{x}_i))^2 + \sum_{i \in \mathcal{O}(A_R)} (y_i - \hat{\psi}_{A_R}(\mathrm{x}_i))^2 \right].$$

At cell A, we compute $R^{\mathrm{SGT}}_{\mathrm{OOB}}(A)$ for the model-based candidate split induced by the chosen $\texttt{hcut}$ value. We also compute $R^{\mathrm{CART}}_{\mathrm{OOB}}(A)$ for the best CART coordinate-threshold split at the same cell. The CART split is learned on the in-bag data and evaluated on the same OOB points. When evaluating the CART candidate, we use the same parent predictor in the first term and replace the daughter predictors by the CART daughter in-bag sample averages. Since the parent term is identical in both calculations, this comparison reduces to which set of daughters yields smaller OOB error. If

$$R_{\mathrm{OOB}}^{\mathrm{CART}}(A) > R_{\mathrm{OOB}}^{\mathrm{SGT}}(A),$$

we use the CART split at A and restrict the entire subtree rooted at A to CART coordinate-threshold splits by setting $\texttt{hcut} = 0$ for all descendants of A. Because this decision is driven by held-out error, it targets generalization performance directly and reduces the optimistic bias of global training searches.

4.2.4 Summary of complexity controls and stabilization

Algorithm 2 controls complexity and encourages stability at multiple levels:

- *Tree size and node size.* K caps the number of splits, and the minimum node-size constraint prevents splitting small cells. BSF is an aggressive *search* strategy, but for fixed $(K, m_{\min})$ it does not expand the tree class beyond trees with at most K splits.
- *Dictionary size.* $\texttt{hcut}$ determines the candidate dictionary in (2), and filtering reduces the effective dimension from p to p_F.
- *Local regularization.* Lasso shrinkage, with a cross-validated penalty, stabilizes split-defining models and induces sparsity.
- *Guard against instability.* The OOB stability rule reverts to CART when model-based splits do not improve held-out performance.

In principle, K can also be selected by cross-validation or by OOB criteria. In the experiments below we report results for fixed values of K specified in each study, since the safeguards above already yielded stable performance.

4.3 Flexibility of cuts

We begin with a controlled, noiseless regression example designed to separate the effect of cut geometry from the effect of the induction strategy. Throughout this subsection, all trees are grown using the same best-split-first (BSF) procedure and the same number of splits K. The only component that changes across fits is the permissible cut family indexed by $\texttt{hcut}$ associated with the local polynomial dictionary. Setting $\texttt{hcut} = 0$ restricts splits to standard coordinate-threshold (CART) cuts while still using BSF. We therefore use $\texttt{hcut} = 0$ as an *axis-parallel BSF baseline* ("best-first CART") for isolating the role of split geometry.

Data are generated from the noiseless nonlinear model

$$Y = \psi(X_1, X_2) = \beta \sin(\pi X_1 X_2), \qquad \beta = 10,$$

with features $X_1, \ldots, X_p \overset{\text{iid}}{\sim} \mathrm{Unif}(0,1)$. We draw $n = 1000$ observations and include $p = 1002$ predictors, consisting of the two signal variables X_1, X_2 and 1000 additional variables independent of Y. Since $\varepsilon \equiv 0$, the response is deterministic given the predictors, so training risk reflects how well the fitted tree approximates ψ on the observed design points. At the same time, because the feature set contains many irrelevant variables, a sufficiently large tree could still overfit through chance correlations. The tree size is fixed and we compare only the effect of changing $\texttt{hcut}$.

The top-left panel of Figure 5 shows the true regression surface $\psi(x_1, x_2)$. The remaining panels show fitted surfaces obtained from the same training sample under different `hcut` values, all grown with $K = 40$ splits and plotted using identical contour levels. With `hcut` $= 0$ (axis-parallel BSF baseline), the fitted surface exhibits the familiar piecewise-constant form induced by coordinate-threshold cuts. Allowing hyperplane cuts (`hcut` $= 1$) yields a smoother approximation. Increasing `hcut` further produces reconstructions that more closely track the nonlinear structure of ψ.

4.4 Efficient risk reduction

We next examine how richer cut families translate into more efficient empirical risk reduction on a standard benchmark. We again keep the induction strategy fixed (BSF) and vary only the permissible cut family through `hcut` .

We use the `friedman1` regression model

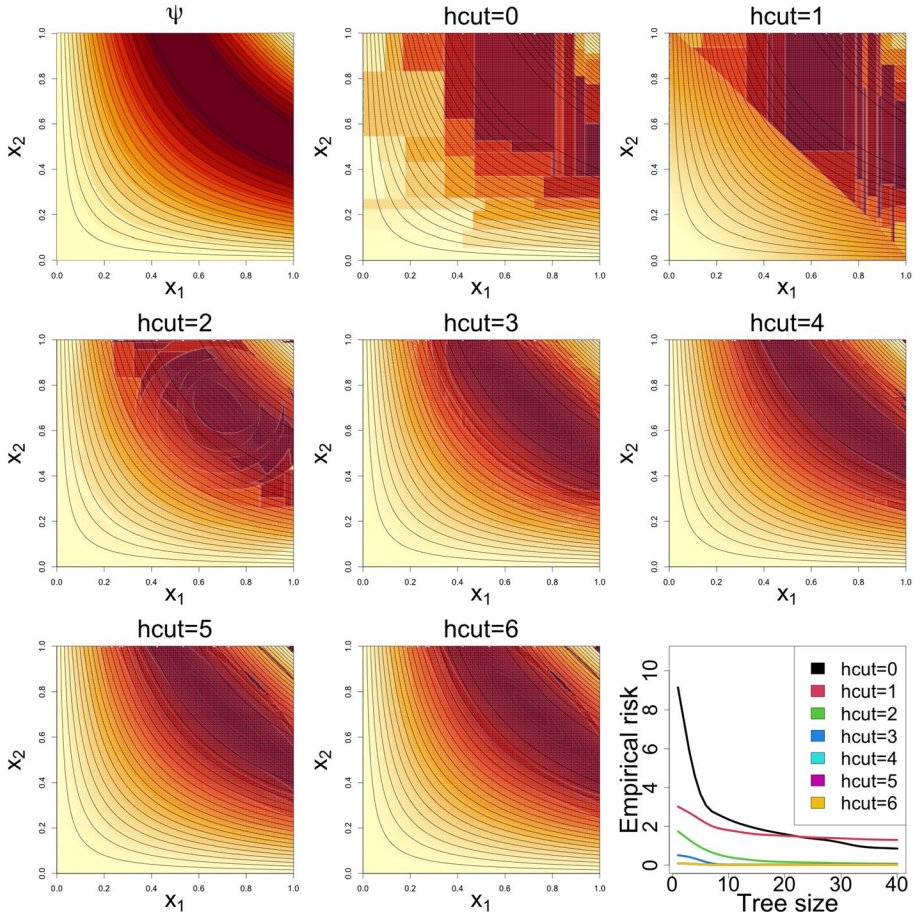


Fig. 5 Noiseless nonlinear regression surface and fitted SGT surfaces under increasing cut-family complexity. All fits use the same BSF growth strategy with $K = 40$ splits; `hcut`$= 0$ is the axis-parallel BSF baseline (best-first CART)

$$Y = \psi(\mathrm{X}) + \varepsilon, \qquad \psi(\mathrm{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5,$$

where $X_j \overset{\text{iid}}{\sim} \text{Unif}(0,1)$ for $j = 1, \ldots, p$ and $\varepsilon \sim N(0,1)$. We set $n = 500$ and $p = 10$ (five signal and five noise variables). We compare the axis-parallel BSF baseline (`hcut` $= 0$) to SGT with richer cuts (`hcut` $= 3$), with tree size fixed at $K = 1, \ldots, 50$ in both cases. Here out-of-bag (OOB) data are used only for evaluation, and the OOB stability guard is disabled.

To track both training and held-out performance along the growth path, each tree is trained on a randomly selected in-bag subset of size $\lfloor 0.632\, n \rfloor$, with the remaining observations used for evaluation and referred to as OOB for consistency with the ensemble setting. Results from 100 independent replications are summarized in Figure 6. The top and bottom panels show in-bag and OOB risk, respectively, as a function of the number of splits (boxplots across replications). The SGT fit (`hcut` $= 3$) attains lower risk than the axis-parallel baseline across splits. This performance is consistent across both training (inbag) and test (OOB) risk evaluations

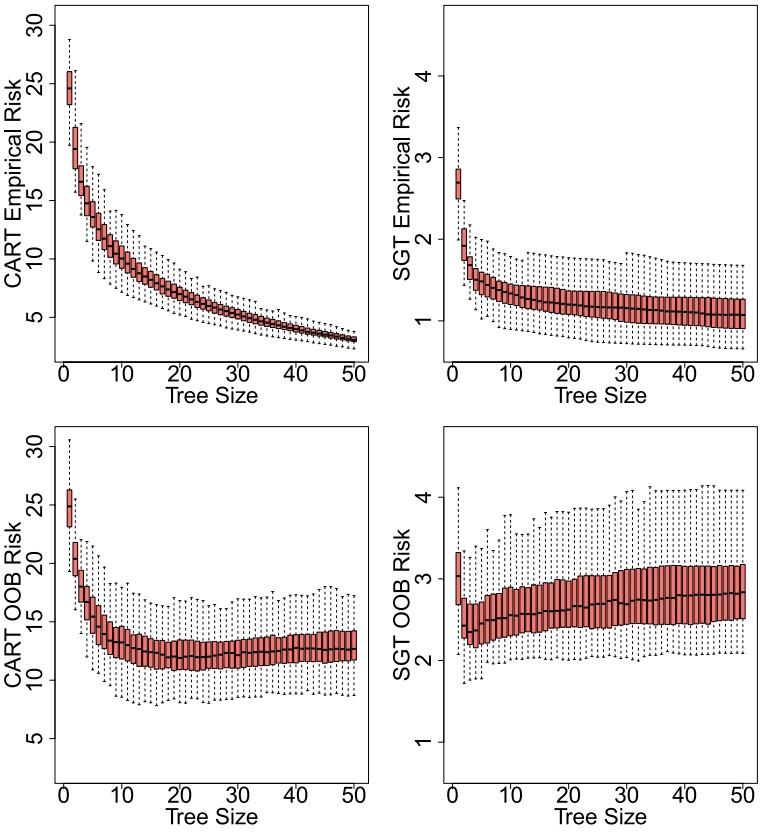


Fig. 6 Empirical risk along the BSF growth path for the `friedman1` simulation. The plot compares the axis-parallel BSF baseline (`hcut`$= 0$) to SGT with richer cuts (`hcut`$= 3$). Boxplots summarize 100 independent runs

4.5 Robustness to "checkerboard"-type problems

We next consider a "checkerboard"-type example adapted from Ferreira (2022), generalizing the construction discussed by Devroye et al. (1996); Biau et al. (2008). The key difficulty is that the response depends on a *joint configuration* of two variables, while neither variable is marginally predictive. In such settings, trees restricted to coordinate-threshold splits can be slow to uncover the relevant structure because candidate splits are driven by univariate associations with the response. This limitation persists even under BSF growth. BSF can prioritize *which* region to split next, but if the cut family cannot express the needed joint feature, many splits may still be required to approximate the target.

We generate data from a latent-switch regression model

$$Y = 1_{\{X_1=X_2\}}\, f(\mathrm{X}) + \big(1 - 1_{\{X_1=X_2\}}\big)\, g(\mathrm{X}) + \varepsilon,$$

where $X_1, X_2 \sim \text{Bernoulli}(1/2)$ independently, $X_3, \ldots, X_{10} \overset{\text{iid}}{\sim} N(0,1)$, and $\varepsilon \sim N(0, 0.1^2)$. The two functions are

$$f(\mathrm{x}) = \sum_{j=3}^{10} \alpha_j x_j^2, \qquad g(\mathrm{x}) = \sum_{j=3}^{10} \beta_j x_j^2,$$

with $\alpha_j = \text{signal} \cdot (j-2)$ for $j = 3, 4$ and $\alpha_j = 0$ otherwise, and $\beta_j = -\alpha_j/2$. We set $\texttt{signal} = 3$. Here X_1 and X_2 act as latent switches. When $X_1 = X_2$, the regression surface is convex in (X_3, X_4), while when $X_1 \neq X_2$ it is concave.

We fit BSF trees using the same number of splits and vary only the cut-family index $\texttt{hcut}$. Contour plots for $\texttt{hcut} \in \{0, 1, 2, 3\}$ are shown in Figure 7 for $n = 5000$. When $\texttt{hcut} = 0$ (axis-parallel BSF baseline), the fitted surfaces show little separation between the $X_1 = X_2$ and $X_1 \neq X_2$ settings. As $\texttt{hcut}$ increases, the fitted surfaces separate these cases more clearly. By $\texttt{hcut} = 3$, which includes pairwise interaction terms that can encode the switching structure, the fitted contours closely track the target pattern. On the $X_1 = X_2$ side, contours concentrate near the edges, reflecting the convex surface, while on the $X_1 \neq X_2$ side the surface peaks near the center, reflecting the concave surface.

4.6 Synthetic benchmark analysis

We evaluated predictive accuracy on 20 synthetic prediction problems that vary in sample size *n*, dimension *p*, and functional form (linear, nonlinear, and mixed). Each method was evaluated on an independently drawn test set, and prediction error was standardized by dividing by the sample variance of the observed response. Each experiment was repeated 100 times.

The data-generating models are listed below.

1. *cobra2*. $\psi(\mathrm{x}) = x_1 x_2 + x_3^2 - x_4 x_7 + x_8 x_{10} - x_6^2$, $X_j \sim U(-1, 1)$, $\varepsilon \sim N(0, 0.1^2)$.
2. *cobra8*. $Y = 1_{\{x_1 + x_4^3 + x_9 + \sin(x_2 x_8) + \varepsilon > 0.38\}}$, $X_j \sim U(-0.25, 1)$, $\varepsilon \sim N(0, 0.1^2)$.
3. *friedman1*. $\psi(\mathrm{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$, $X_j \sim U(0, 1)$, $\varepsilon \sim N(0, 1)$.

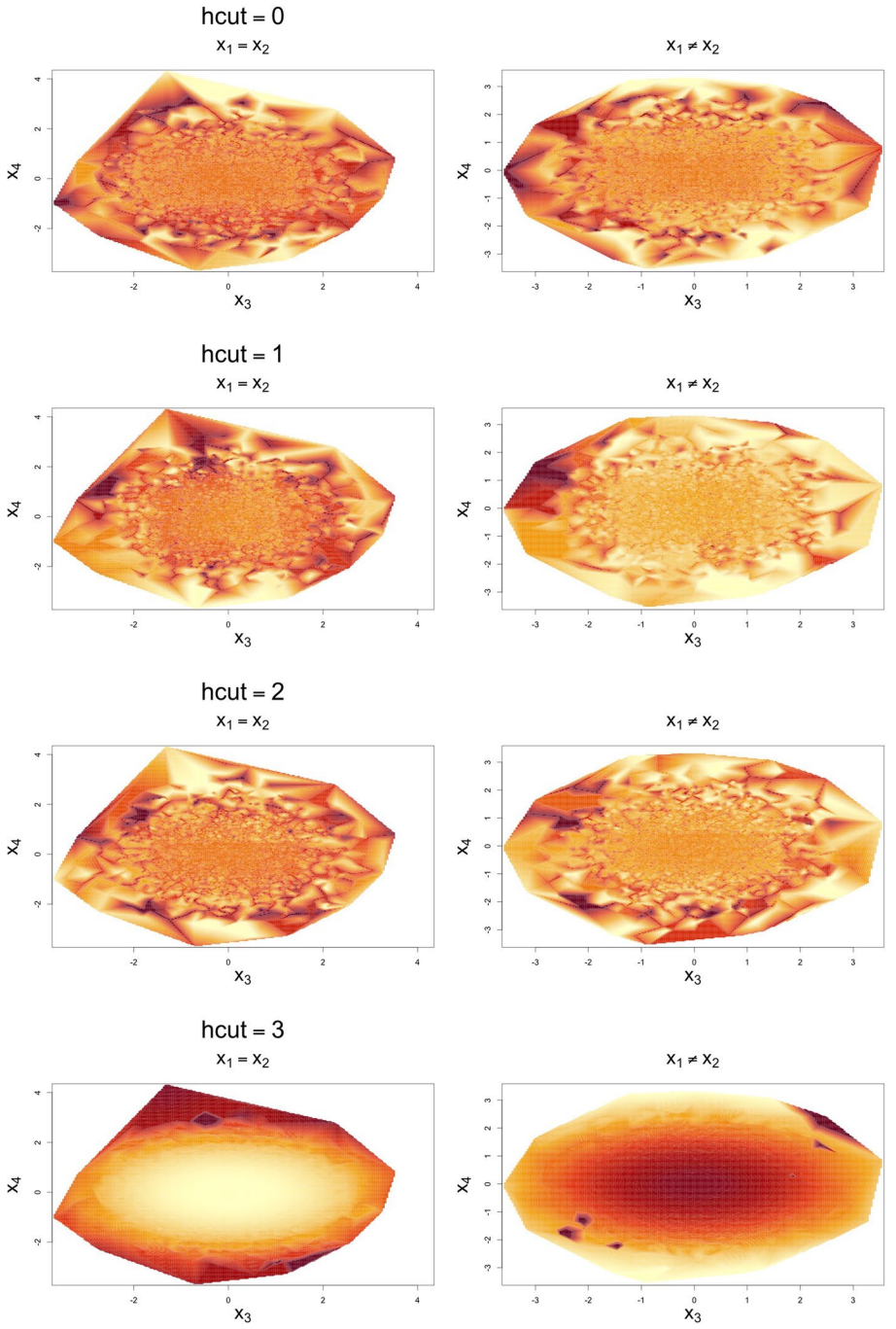


Fig. 7 Estimated regression surfaces under increasing `hcut` for the switching latent-variable simulation. Each panel shows a contour plot over the x_3–x_4 plane, conditioned on fixed values of x_1 and x_2. All fits use the same BSF growth strategy; `hcut`$= 0$ is the axis-parallel BSF baseline

4. *friedman2.* $\psi(\mathrm{x}) = \sqrt{x_1^2 + \left(x_2 x_3 - \frac{1}{x_2 x_4}\right)^2}$, $X_1 \sim U(0,100)$, $X_2 \sim U(40\pi, 560\pi)$, $X_3 \sim U(0,1)$, $X_4 \sim U(1,11)$, $X_5, \ldots, X_p \sim U(0,1)$, $\varepsilon \sim N(0, 95^2)$.
5. *friedman3.* $\psi(\mathrm{x}) = \arctan\left(\frac{x_2 x_3 - \frac{1}{x_2 x_4}}{x_1}\right)$, $X_1 \sim U(0,100)$, $X_2 \sim U(40\pi, 560\pi)$, $X_3 \sim U(0,1)$, $X_4 \sim U(1,11)$, $X_5, \ldots, X_p \sim U(0,1)$, $\varepsilon \sim N(0, 0.1^2)$.
6. *inl1.* $\psi(\mathrm{x}) = x_1 x_2^2 \sqrt{|x_3|} + \lfloor x_4 - x_5 x_6 \rfloor$, $X_j \sim U(-1,1)$, $\varepsilon \sim N(0, 0.1^2)$.
7. *inl2.* $\psi(\mathrm{x}) = x_3 (x_1 + 1)^{|x_2|} - \sqrt{\frac{x_5^2}{|x_4| + |x_5| + |x_6|}}$, $X_j \sim U(-1,1)$, $\varepsilon \sim N(0, 0.1^2)$.
8. *inl3.* $\psi(\mathrm{x}) = \cos(x_1 - x_2) + \arcsin(x_1 x_3) - \arctan(x_2 - x_3^2)$, $X_j \sim U(-1,1)$, $\varepsilon \sim N(0, 0.1^2)$.
9. *lm1.* $\psi(\mathrm{x}) = \sum_{j=1}^{15} x_j$, $X_j \sim N(0,1)$, $\varepsilon \sim N(0, 10^2)$.
10. *lm2.* $\psi(\mathrm{x}) = 2\sum_{j=1}^{15} x_j$, $X_j \sim N(0,1)$, $\varepsilon \sim N(0, 10^2)$.
11. *lm3.* $\psi(\mathrm{x}) = 2\sum_{j=1}^{15} x_j$, $X_j \sim N(0,1)$, $\varepsilon \sim N(0, 5^2)$.
12. *lmi1.* $\psi(\mathrm{x}) = 0.05 f_1(\mathrm{x}) + \exp(0.02 f_1(\mathrm{x}) f_2(\mathrm{x}))$, where $f_1(\mathrm{x}) = \sum_{j=1}^{10} x_j$, $f_2(\mathrm{x}) = \sum_{j=11}^{20} x_j$, $X_j \sim U(0,1)$, $\varepsilon \sim N(0, 0.05^2)$.
13. *lmi2.* $\psi(\mathrm{x}) = 3(\sum_{j=1}^{15} x_j)^2$, $X_j \sim N(0,1)$, $\varepsilon \sim N(0,1)$.
14. *peak.* $\psi(\mathrm{x}) = 25\exp(-0.5r^2)$, $r \sim U(0,3)$, $Z_1, \ldots, Z_p \sim N(0,1)$, $X_j = \frac{r Z_j}{\sqrt{\sum_{j=1}^{p} Z_j^2}}$.
15. *sup.* $\psi(\mathrm{x}) = 10 x_1 x_2 + \frac{0.25}{x_3 x_4 + 10 x_5 x_6}$, $X_j \sim U(0.05, 1)$, $\varepsilon \sim N(0, 0.5^2)$.
16. *sup2.* $\psi(\mathrm{x}) = \pi^{x_1 x_2}\sqrt{2x_3} - \arcsin(x_4) + \log(x_3 + x_5) - \frac{x_9}{x_{10}}\sqrt{\frac{x_7}{x_8}} - x_2 x_7$, $X_j \sim U(0.5, 1)$.
17. *xsup.* Same as *sup* but with smaller *n* and larger *p*.
18. *xsup2.* Same as *sup2* but with smaller *n* and larger *p*.

Simulations *cobra2* and *cobra8* are from Biau et al. (2016), and *friedman1*, *friedman2*, and *friedman3* are from Friedman (1991). Unless noted otherwise, we set $n = 1000$ and $p = 20$, except for *xsup* and *xsup2*, where $n = 500$ and $p = 200$. For models in which ε is not specified above, we set $\varepsilon \equiv 0$.

In addition, we used the function `regDataGen` from the R package `CORElearn`. In these experiments the regression function switches between two settings,

$$\psi(\mathrm{x}) = x_4 - 2x_5 + 3x_6 \qquad \text{or} \qquad \psi(\mathrm{x}) = \cos(4\pi x_4)(2x_5 - 3x_6),$$

with the choice determined by a latent switch variable. The simulation includes four discrete variables a_1, a_2, a_3, a_4, where a_1 and a_2 are informative about the switch, and continuous variables X_1 and X_2 that also carry information about the latent setting. We modified the

generator to allow additional noise variables distributed as $U(0, 1)$. We considered the following two high-dimensional settings.

19. *corelearn1.* a_1, a_2 and X_1, X_2 contain full information, $n = 100$, and 50 noise variables are added.
20. *corelearn2.* a_1, a_2 and X_1, X_2 contain full information, $n = 100$, and 200 noise variables are added.

4.6.1 Super greedy forests

To reduce variance while retaining the flexibility of deep trees, we use ensembles. We draw 100 subsamples of size $\lfloor 0.632\, n \rfloor$ without replacement and fit one tree per subsample. Predictions are averaged to form a *Super Greedy Forest* (SGF). We write SGF-*h* for a forest grown with `hcut`$= h$ (for example, SGF-3 uses `hcut`$= 3$). We set $K = 200$ for SGF-0 and $K = 100$ for all other SGFs. SGF-0 uses CART-style axis-parallel splits with random feature selection, and it represents a best-split-first analogue of random forests (RF). We also define SGF-opt as the SGF whose `hcut` value is chosen to minimize 10-fold cross-validation error. For all forests with `hcut`> 0, we enable the OOB stability guard described in Section 4.2.

4.6.2 Comparison procedures

Comparison procedures include generalized boosted trees (GBT) (Friedman 2001) using the R package `gbm` (Greenwell et al. 2020), with up to 5000 trees tuned via 10-fold cross-validation. We also tested local linear forests (LLF) (Friedberg et al. 2020) using the R package `gfr` (Tibshirani et al. 2022), with ridge regularization and an option that performs lasso-based feature selection (LLF-lasso), following the implementation guidelines in https://grf-labs.github.io/grf/articles/llf.html. Oblique random forests (ORF) were compared using the R package `ODRF` (Liu and Xia 2022), which optimizes linear combinations of features for splits. Finally, we include standard random forests (RF). RF is closest in spirit to SGF-0, but differences in induction and preprocessing motivate its separate inclusion.

4.6.3 Results

Standardized prediction errors for all 20 benchmark experiments are shown in Figure 8. Overall performance is summarized in Figure 9 using a critical difference (CD) plot (Demšar 2006), which reports the average rank of each method across datasets (lower is better) and connects groups that are not significantly different at the 5% level.

- *Overall.* The CD plot in Figure 9 shows that SGFs with richer cut families tend to perform best on average. SGF-3 through SGF-5 and SGF-opt form the leading group, and SGF-2 is close behind with a slightly larger average rank. GBT follows next. RF, ORF, and SGF-0 have larger average ranks, while LLF generally improves on these methods and LLF-lasso improves further in several settings.
- *Linear models.* In the linear experiments (*lm1*, *lm2*, *lm3*), methods with hyperplane or richer cuts outperform RF and SGF-0, which is consistent with known limitations of

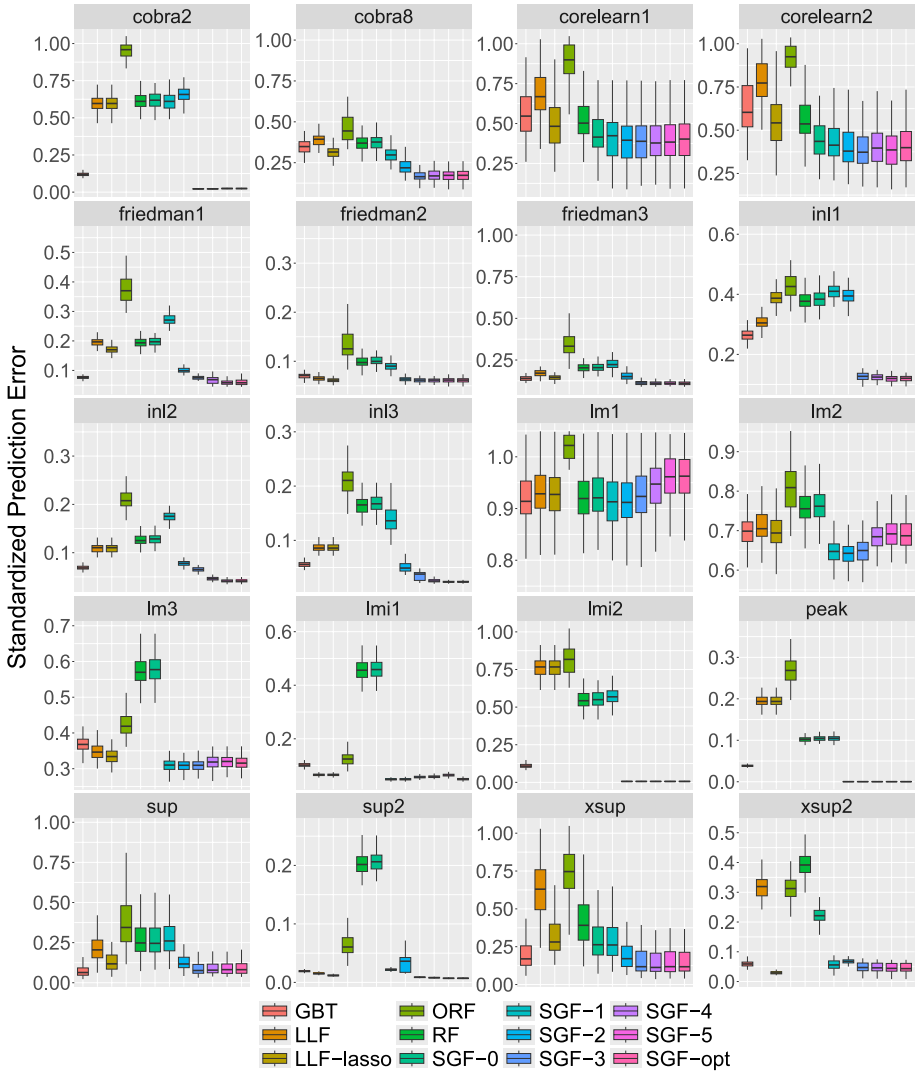


Fig. 8 Standardized prediction error in the synthetic benchmark study

coordinate-threshold forests on linear signals.

- *Quadratic terms and interactions.* In models with strong quadratic and interaction structure (*cobra2*, *lmi1*, *lmi2*), RF performs poorly. In the most challenging cases (for example, *cobra2* and *lmi2*), the best performance is concentrated among GBT and SGF-2 through SGF-5 (including SGF-opt).
- *Nonlinear models.* SGFs, especially SGF-3 through SGF-5 and SGF-opt, perform well across nonlinear settings such as *friedman1–3* and *inl1–3*. This demonstrates the ability of node-adaptive geometric cuts to approximate nonlinear structure even when split proposals come from a parametric family.
- *Larger p and smaller n*. In high-dimensional, small-sample experiments (*corelearn1*,

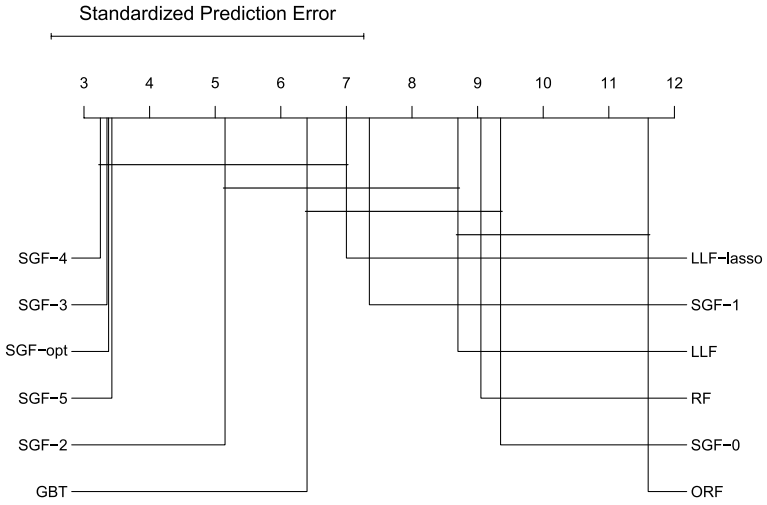


Fig. 9 Critical difference (CD) plot summarizing the average rank of each procedure across the 20 benchmark experiments in Figure 8. Lower ranks indicate better performance. Horizontal bars connect methods that are not significantly different at the 5% level

corelearn2, *xsup*, *xsup2*), methods with built-in regularization or dimensionality reduction (GBT, LLF-lasso, and SGFs with larger `hcut`) perform best.

- *Hyperplane cuts.* SGF-1 (hyperplane cuts) performs similarly to LLF and improves upon ORF, while LLF-lasso improves upon both.
- *SGF-0 and RF.* SGF-0 generally matches, and sometimes modestly improves upon, RF. The differences are more visible in the higher-dimensional settings (*xsup*, *xsup2*), where SGF-0 benefits from its filtering and split-selection implementation.
- *Choosing* `hcut`. Increasing `hcut` improves SGF performance in most settings, although very rich dictionaries can overfit in simpler problems. SGF-opt, which selects `hcut` by cross-validation, remains consistently competitive across datasets.

4.7 Evaluation on diverse benchmark datasets

To evaluate SGFs under diverse conditions, we selected regression problems from the Penn Machine Learning Benchmark (PMLB) repository (Olson et al. 2017; Palaniappan et al. 2025). After filtering for datasets with at most 2000 observations, at least 10 input features, and continuous outcomes, we retained 61 datasets.

Based on the synthetic benchmark, we compared SGF, RF, GBT, and LLF. For SGF we used SGF-opt, selecting `hcut` by cross-validation. Performance was evaluated using 10-fold cross-validation, with prediction error standardized by the sample variance of Y. Each experiment was repeated 25 times independently.

Average ranks across datasets are summarized in n Figure 10 using a CD plot, and the corresponding performance values are shown in Figure 11. SGF and GBT achieve the best overall performance and are not significantly different at the 5% level, with SGF having a slightly smaller average rank. RF and LLF form a lower-performing group, with RF outper-

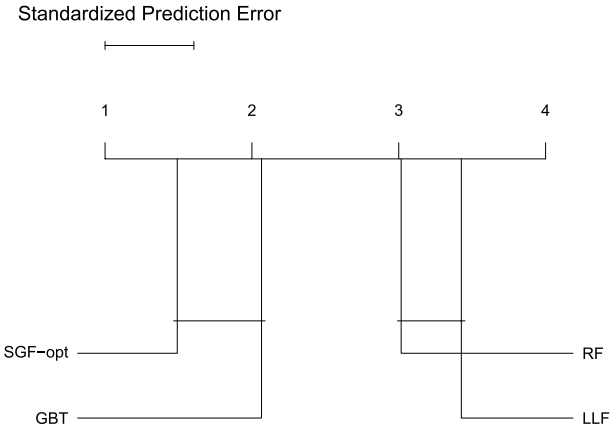


Fig. 10 Critical difference (CD) plot showing the average rank of each method across 61 regression datasets from PMLB. Each experiment was repeated 25 times using 10-fold cross-validation. Lower ranks indicate better performance. Horizontal bars connect methods that are not significantly different at the 5% level

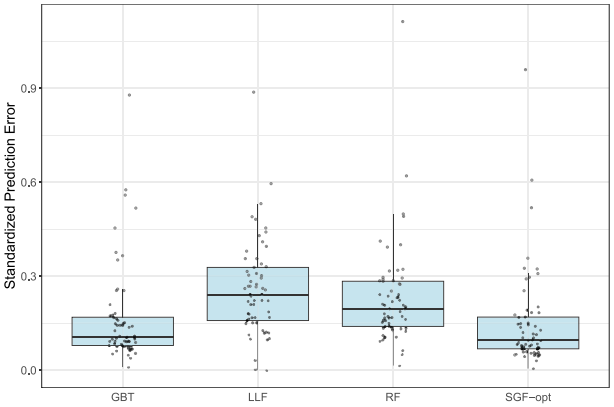


Fig. 11 Boxplots of standardized prediction error across datasets in the PMLB benchmark study

forming LLF. Overall, SGF performs on par with boosted trees across this diverse collection of benchmark regression datasets.

5 Parametric and nonparametric insights in data analysis

In addition to predictive accuracy, the SGT framework provides interpretable parametric summaries of the fitted tree predictor. We use this idea in two specific ways. First, each split and each terminal-node predictor is defined by a sparse lasso-fit parametric model, so only a small set of terms is active within any node. The variables and interactions that matter in that region can therefore be seen directly from the nonzero coefficients. Second, for any covariate vector x, the fitted value can be decomposed into a sum of term-level contributions from individual variables and interactions. This gives an observation-level breakdown of how the prediction is assembled, which helps summarize associations between features and outcome.

These same local summaries extend naturally to ensembles. For an ensemble (SGF), we aggregate the nodewise coefficient information across trees, which yields summary values

with the added benefit of being more stable. We illustrate these ideas using a clinical case study and synthetic benchmarks.

5.1 Predicting mortality using treadmill ECG and clinical data

We analyzed a clinical cohort of patients who underwent treadmill exercise testing for evaluation of suspected coronary artery disease. All patients had a clinically normal resting electrocardiogram (ECG) and no prior history of cardiovascular disease at the time of testing. The dataset, previously analyzed in Gorodeski et al. (2009), includes $n = 18{,}964$ patients and more than $p = 150$ clinical and ECG-derived features.

The primary outcome was all-cause mortality. A total of 1,585 deaths (8%) occurred over a median follow-up of 10.7 years, with follow-up ranging from 5 to 17 years. To obtain a continuous response suitable for SGT regression, we first fit random survival forests (RSF) (Ishwaran et al. 2008) to estimate patient-level survival functions. We then define the regression target y as the restricted mean survival time (RMST) (Irwin 1949; Andersen et al. 2004; Royston and Parmar 2011; Kim et al. 2017) up to a fixed horizon $\tau > 0$,

$$\mathrm{RMST}(\tau) = \int_0^\tau S(t)\, dt,$$

where $S(t)$ is the estimated survival probability at time t. We set $\tau = 10$ years, a clinically meaningful horizon for long-term survival in this population.[1]

The RSF model used all available features. Clinical variables included demographics and medical history (e.g., sex, diabetes, hypertension, smoking) as well as exercise-related metrics such as heart rate recovery and exercise capacity. ECG-derived variables captured repolarization and conduction features, heart-rate variability measures, and proxies for left ventricular mass.

5.1.1 Coefficient functions and term contributions.

Our goal is to identify and quantify predictors of RMST using SGF coefficients. For a forest with B trees, let $\hat{\beta}^{(b)}(\mathrm{x})$ denote the coefficient vector of the terminal-node model in tree b that contains x (that is, the lasso coefficient vector for the leaf containing x in tree b). Define the (coordinate-wise) *coefficient functions* $\hat{\beta}(\mathrm{x}) = B^{-1} \sum_{b=1}^{B} \hat{\beta}^{(b)}(\mathrm{x})$. The ensemble predictor can then be written as a parametric expansion with x-dependent coefficients. To facilitate interpretation, we restrict attention to intercepts, linear terms, and pairwise interactions and write

$$\hat{\psi}(\mathrm{x}) = \hat{\beta}_0(\mathrm{x}) + \sum_{l=1}^{p} \hat{\beta}_l(\mathrm{x})\, x_l + \sum_{1 \le l_1 < l_2 \le p} \hat{\beta}_{l_1 l_2}(\mathrm{x})\, x_{l_1} x_{l_2}.$$

[1] This two-stage construction can be viewed as a form of distillation. RSF provides a flexible survival estimate, while SGF provides a term-by-term parametric decomposition of the resulting RMST predictions.

This representation yields a natural additive decomposition of the fitted value into term-level contributions. We define the *linear-term contribution* of variable x_l at covariate vector x by

$$\hat{\theta}_l(\mathrm{x}) = \hat{\beta}_l(\mathrm{x})\, x_l.$$

For an interaction pair (l_1, l_2) we define the *interaction-term contribution* as $\hat{\beta}_{l_1 l_2}(\mathrm{x}) x_{l_1} x_{l_2}$. When summarizing a pair, it is useful to report the combined (main + interaction) contribution

$$\hat{\theta}_{l_1 l_2}(\mathrm{x}) = \hat{\beta}_{l_1}(\mathrm{x}) x_{l_1} + \hat{\beta}_{l_2}(\mathrm{x}) x_{l_2} + \hat{\beta}_{l_1 l_2}(\mathrm{x}) x_{l_1} x_{l_2},$$

which summarizes the joint contribution of the pair at x.

5.1.2 Linear contributions.

Figure 12 shows patient-level linear-term contributions for selected variables. Among clinical variables, peak metabolic equivalents (peak_mets) has the strongest positive contribution to RMST, consistent with longer predicted survival among fitter individuals. Several ECG features show negative contributions, including T-wave amplitude in lead I (tamp_I) and lead aVL (tamp_aVL), and the ST-segment end in lead I (ste_I), suggesting that repolarization differences are associated with lower predicted RMST in this cohort. The Romhilt–Estes score (RE_Score), a proxy for left ventricular mass, is also negatively associated with predicted RMST. Other effects (e.g., timing and slope measures in lateral leads) are comparatively modest. It is noteworthy, after accounting for the other terms selected by the forest, variables such as sex and Duke score have small marginal linear contributions in this model.

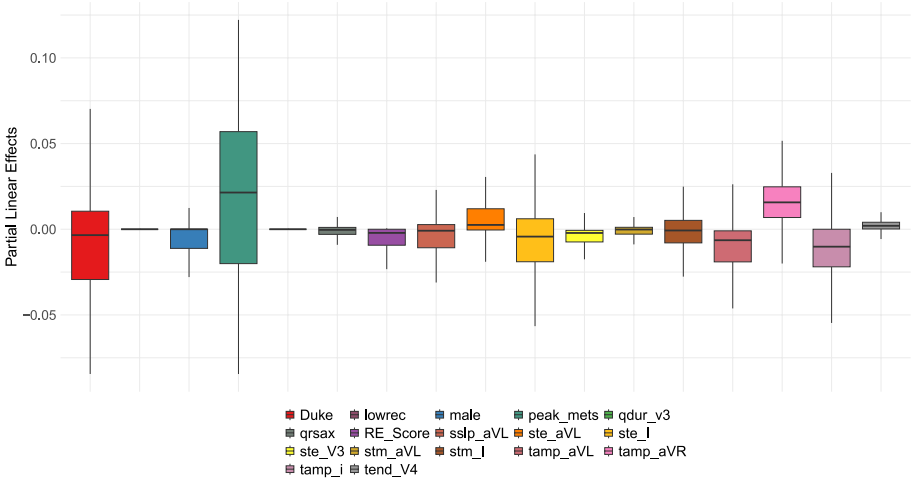


Fig. 12 Patient-level linear-term contributions for RMST prediction. Negative values correspond to lower predicted RMST (shorter expected survival)

5.1.3 Interaction contributions.

Figure 13 displays several prominent pairwise joint contributions. A large positive joint contribution appears for the interaction between Duke score and peak metabolic equivalents. Additional interactions link ECG repolarization features with exercise capacity, including sslp_aVL × peak_mets, tamp_aVL × peak_mets, and ste_aVL × peak_mets, each corresponding to larger predicted RMST. These patterns indicate that higher RMST predictions tend to occur when exercise capacity is high and repolarization features are favorable.

Among clinical–ECG pairs, the largest positive joint contribution involves exercise capacity (peak_mets) and ST-segment slope in lead aVL (sslp_aVL). To examine this association, we stratified RMST across joint bins of these two variables. As shown in Figure 14, RMST increases consistently with higher exercise capacity, and the gradient is stronger when ST slope is more upright. For example, when sslp_aVL $\in (0.5, 43]$, median RMST increases from approximately 9.3 years at low exercise capacity to nearly 9.9 years for peak_mets > 13. Similar patterns appear across other slope strata, which is consistent with the joint contribution pattern in the forest.

5.2 Synthetic experiments demonstrating parametric insights

As a second illustration, we applied the same coefficient-function and contribution summaries to three synthetic benchmarks: *friedman1*, *peak*, and *lmi2*. Figure 15 shows the distribution of partial contributions estimated by SGF with `hcut` $= 3$ across 250 independent runs ($n = 1000, p = 20$). These contributions summarize how much each term contributes to the predicted response tree averaged over the local model structures.

In *friedman1*, the largest contributions arise from terms involving x_1 and x_2 (indicating interactions), together with strong effects from x_3 (quadratic behavior) and x_4 and x_5 (additive linear components), which is consistent with the data-generating mechanism. In *peak*, the values concentrate on quadratic terms while linear terms are negligible, reflecting the

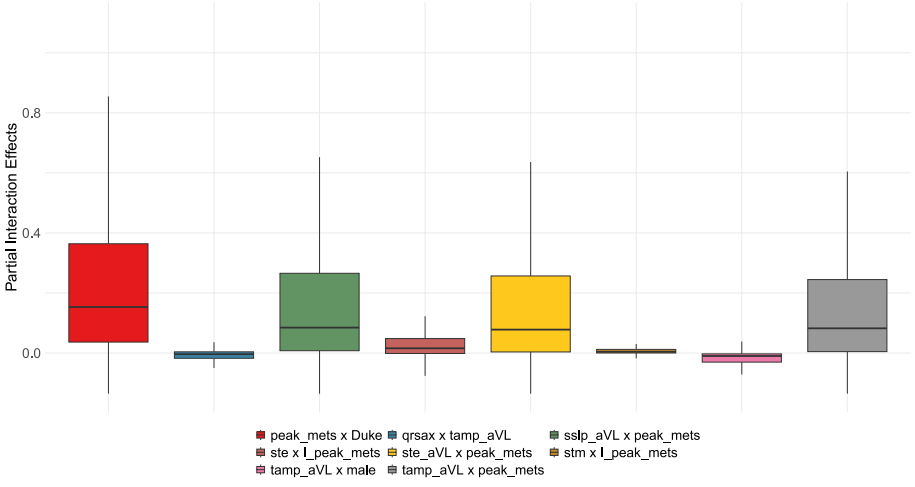


Fig. 13 Patient-level pairwise joint contributions for RMST prediction. Interactions summarize multivariate combinations of ECG and clinical variables associated with differential predicted survival

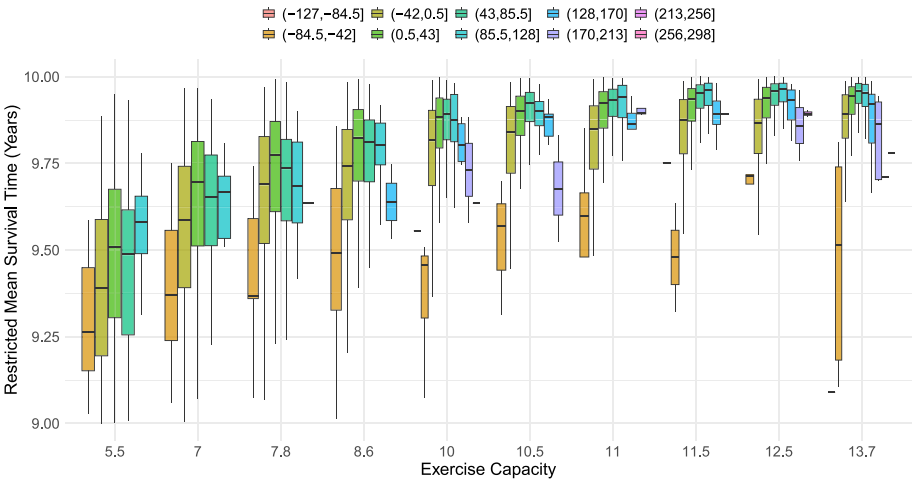


Fig. 14 RMST (years) as a function of exercise capacity, stratified by bins of ST-segment slope in lead aVL (sslp_aVL). RMST increases with both higher exercise capacity and more upright ST slope, indicating a joint association with longer predicted survival

radial symmetry of the signal. In *lmi2*, the signal is globally quadratic, and the fitted contributions are spread across quadratic and interaction terms among the active variables, matching the squared-sum structure of the target.

6 Discussion

6.1 Summary

We introduced Super Greedy Trees (SGTs), a decision-tree framework that uses a class of sparse parametric score functions to define multivariate, greedy geometric cuts. Local models are fit with the lasso, which adapts the effective complexity of each cut to the data available in the node. Near the root, where sample sizes are larger, the fitted score can support richer boundaries (for example, quadratic forms). Deeper in the tree, lasso sparsity and smaller node sizes tend to favor simpler split rules, often approaching traditional axis-parallel (coordinate-threshold) cuts.

Section 4 evaluated Super Greedy Forests (SGFs) against random forests, gradient boosted trees (GBT), and oblique random forests (ORF) across two benchmark studies. SGFs performed well overall, with best performance obtained when cut complexity, controlled by `hcut`, was selected adaptively. Theorem 1 helps clarify why this can happen. Enlarging the underlying cut class increases the induced partition space, as measured by shatter-coefficient and partitioning-number bounds. Section 3.5 then shows that the resulting increase in estimation error remains controlled when the number of splits and the within-node model complexity are limited. Under this control, richer cut families can lower approximation error and improve test performance. This confirms what we found in Section 4.

A second strength of the framework is that it combines parametric and nonparametric modeling while retaining interpretability. The lasso fits used for splitting and for leaf

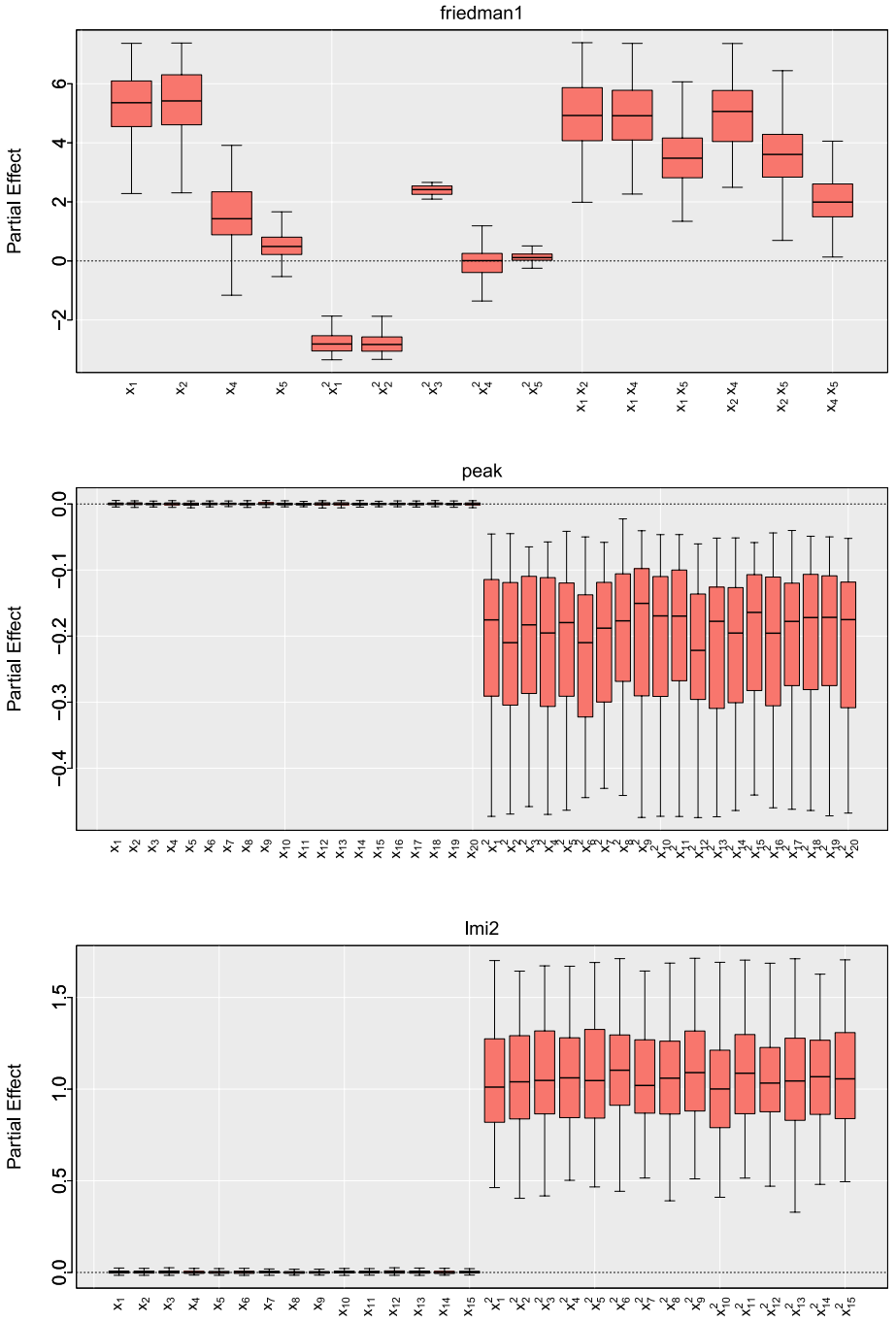


Fig. 15 SGF partial contributions from synthetic experiments using 250 independent runs ($n = 1000$, $p = 20$). Panels correspond to *friedman1* (top), *peak* (middle), and *lmi2* (bottom). Partial contributions represent the aggregated term-level contribution of each variable or interaction to the predicted response

prediction yield sparse local models, so each region is described by a small set of active variables and interactions. Section 5 showed how these local fits can be aggregated in SGFs to produce observation-specific term contributions (linear and interaction) that summarize predictive associations. In the treadmill ECG case study, these summaries revealed clinically meaningful patterns linked to long-term survival, including interactions between ECG morphology and exercise physiology. The synthetic experiments further showed that SGFs can recover structured parametric forms, such as interaction signals and multivariate quadratic structure.

6.2 Computational complexity

These benefits come with computational costs. Multivariate cuts require fitting local parametric models, and the size of the candidate dictionary grows polynomially with the effective dimension. We addressed this using feature filtering driven by lasso sparsity, which reduces the original dimension to a smaller set.

To quantify complexity for the implementation used in Section 4, let n be the sample size and p the original dimension. After the two-stage filtering step, let p_F denote the number of retained covariates, and let $d(\texttt{hcut}, p_F)$ be the number of candidate basis terms for the parametric model induced by `hcut`. For example, $d(1, p_F) = 1 + p_F$, while $d(3, p_F) = 1 + 2p_F + \binom{p_F}{2} = O(p_F^2)$. Let K be the number of splits in the final tree (so there are $K + 1$ terminal nodes), and let D denote the resulting tree depth (with $D \le K$).

In practice, BSF growth is implemented with a priority queue so that split candidates for unchanged terminal nodes are not recomputed. Each time a node is split, only the two newly created terminal nodes are evaluated. Therefore the total number of node evaluations is $1 + 2K$.

Consider a terminal node A containing $M = M(A)$ in-bag observations. Evaluating the SGT split rule at A involves three steps. First, fit a lasso model on A. Second, sort fitted values and scan thresholds to minimize (3). Third, refit lasso models on the two resulting daughters to compute the empirical risk reduction. Let C_{lasso} denote the constant associated with fitting one lasso model per observation per candidate term. This constant includes coordinate-descent iterations, the regularization path, and the 10-fold cross-validation used to select the penalty. Treating the constant number of lasso refits as part of C_{lasso}, the cost of one node evaluation is

$$O\Big(C_{\text{lasso}}\, M\, d(\texttt{hcut}, p_F) + M \log M\Big).$$

Summing over the $1 + 2K$ evaluated nodes yields the training cost

$$T_{\text{SGT}} = O\Big(C_{\text{lasso}}\, d(\texttt{hcut}, p_F) \sum_A M(A) + \sum_A M(A) \log M(A)\Big), \qquad \sum_A 1 = 1 + 2K,$$

where the sums run over evaluated nodes. Since each observation belongs to at most $D + 1$ nodes along its root-to-leaf path, $\sum_A M(A) \le n(D + 1) = O(nD)$. Also, since $M(A) \le n$, $\sum_A M(A) \log M(A) = O(nD \log n)$. Therefore,

$$T_{\mathrm{SGT}} = O\Big(C_{\mathrm{lasso}}\, d(\mathrm{hcut}, p_F)\, nD + nD \log n\Big).$$

For comparison, consider a CART-based random forest with B trees, each grown to K splits with depth D. At a node with M observations, a standard CART split searches over `mtry` candidate features. The per-node cost is $O(\mathrm{mtry}\, M)$ given presorting-based preprocessing, or $O(\mathrm{mtry}\, M \log M)$ if one accounts for nodewise sorting. Aggregating yields

$$T_{\mathrm{RF}} = O\Big(B\, \mathrm{mtry}\, nD\Big) \quad \text{or} \quad T_{\mathrm{RF}} = O\big(B\, \mathrm{mtry}\, nD \log n\big) \text{ with nodewise sorting.}$$

The SGF with B trees has training time on the order of $B\, T_{\mathrm{SGT}}$, plus the overhead of filtering and any `hcut` selection. Thus relative to RF, SGFs replace the `mtry` univariate split search with lasso fitting, where the dominant term is $d(\mathrm{hcut}, p_F)$ and the constant C_{lasso}. Meanwhile the effect of n and the depth factor remain the same. This makes clear that the dimension reduction step that reduces p to p_F is important for keeping SGF computations close to RF in high-dimensional settings.

6.3 Future work

Several extensions are natural. First, the framework can be adapted to other response types (classification, count outcomes, and general GLM losses) and to direct survival objectives. For example, one can fit splits and leaf models using censored survival data directly rather than relying on a two-stage RMST construction. Second, richer cut dictionaries beyond polynomial terms, including splines, structured interactions, and domain-specific feature expansions, may improve accuracy while still producing sparse local summaries. Third, there is room for computational improvements, including warm-start strategies along the lasso path, screening rules for the candidate dictionary, and parallelization across nodes or trees. Finally, while this paper emphasized predictive performance, the case study in Section 5 points to another important direction. It motivates scalable summaries for understanding how variables and outcomes are related, especially when interactions and nonlinear effects play a substantial role.

Author Contributions H.I. is the sole author of the paper, all work, including theoretical, computation and practical, were carried about H.I.

Funding Research for the author was supported by the National Institute Of General Medical Sciences of the National Institutes of Health, Award Number R35 GM139659 and the National Heart, Lung, and Blood Institute of the National Institutes of Health, Award Number R01 HL164405.

Data Availability The Penn Machine Learning Benchmark (PMLB) repository (Olson et al., 2017; Palaniappan et al., 2025) was used for one of our regression benchmark studies. All datasets are publicly available and accessible through the pmlbr R package. As described in the main text, 61 datasets were selected based on specific filtering criteria to ensure diversity and comparability.

Code Availability Our method is implemented in the R package `randomForestSGT`, available at https://github.com/kogalur/randomForestSGT. While benchmark scripts are not publicly posted, all simulations and real-world experiments are described in detail in the manuscript to support reproducibility. Additional documentation and user vignettes are available at https://www.randomforestsgt.org.

Declarations

Conflict of interest The author declares no conflict of interest.

References

Andersen PK, Hansen MG, Klein JP (2004) Regression analysis of restricted mean survival time based on pseudo-observations. Lifetime Data Anal 10(4):335–350

Avellaneda F (2025), Learning optimal oblique decision trees with (Max)SAT, *in* J. Kwok, ed., Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25, International Joint Conferences on Artificial Intelligence Organization, pp. 2558–2565

Bertsimas D, Dunn J (2017) Optimal classification trees. Mach Learn 106:1039–1082

Bertsimas D, Dunn J, Wang Y (2021) Near-optimal nonlinear regression trees. Oper Res Lett 49(2):201–206

Biau G, Devroye L, Lugosi G (2008), 'Consistency of random forests and other averaging classifiers', *J Machine Learn Res* 9(Sep), 2015–2033

Biau G, Fischer A, Guedj B, Malley JD (2016) COBRA: A combined regression strategy. J Multivar Anal 146:18–28

Blaser R, Fryzlewicz P (2016) Random rotation ensembles. J Machine Learn Res 17(1):126–151

Blum R, Hiabu M, Mammen E, Meyer JT (2024) Consistency of random forest type algorithms under a probabilistic impurity decrease condition, arXiv preprint arXiv:2309.01460 . Version 2, last revised 20 Feb 2024

Blum R, Hiabu M, Mammen E, Meyer JT (2025) Pure interaction effects unseen by random forests. Comput Stat Data Anal 212:108237

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and Regression Trees. CRC Press

Brodley CE, Utgoff PE (1995) Multivariate decision trees. Mach Learn 19(1):45–77

Bruckner A (1962) Tests for the superadditivity of functions. Proceedings of the American Mathematical Society 13(1):126–130

Carreira-Perpiñán MÁ, Tavallali P (2018) Alternating optimization of decision trees, with application to learning sparse oblique trees, *in* Advances in Neural Information Processing Systems, Vol. 31

Carta A, Frigau L (2025) Tree oblique for regression with weighted support vector machine. Comput Stat 40:5257–5291

Cattaneo MD, Chandak R, Klusowski JM (2024) Convergence rates of oblique regression trees for flexible function libraries. Ann Stat 52(2):466–490

Chan K-Y, Loh W-Y (2004) LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. J Comput Graph Stat 13(4):826–852

Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system, *in* International Conference on Knowledge Discovery and Data Mining, pp. 785–794

Chipman HA, George EI, McCulloch RE (2010) Bart: Bayesian additive regression trees. Ann Appl Stat 4(1):266–298. https://doi.org/10.1214/09-AOAS285

Cover TM (1965) Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans Electron Comput 3:326–334

da Silva N, Cook D, Lee E-K (2021) A projection pursuit forest algorithm for supervised classification. J Comput Graph Stat 30(4):1168–1180

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
Devroye L, Györfi L, Lugosi G (1996) *A Probabilistic Theory of Pattern Recognition*, Springer
Dudley RM (1978) Central limit theorems for empirical measures. Ann Probab 6(6):899–929
Ferreira JA (2022) Models under which random forests perform badly; consequences for applications. Comput Stat 37(4):1839–1854
Frank E, Wang Y, Inglis S, Holmes G, Witten IH (1998) Using model trees for classification. Mach Learn 32(1):63–76
Frank E, Witten IH (1998) Generating accurate rule sets without global optimization, *in* Proc. 15th International Conference on Machine Learning, Morgan Kaufmann, pp. 144–151
Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm, *in* Proc. 13th International Conference on Machine Learning, Morgan Kaufmann, pp. 148–156
Friedberg R, Tibshirani J, Athey S, Wager S (2020) Local linear forests. J Comput Graph Stat 30(2):503–517
Friedman JH (1991) Multivariate adaptive regression splines. Ann Stat 19(1):1–67
Friedman JH (2001) Greedy function approximation: A gradient boosting machine. Ann Stat 29:1189–1232
Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1–22
Gorodeski EZ, Ishwaran H, Blackstone EH, Lauer MS (2009) Quantitative electrocardiographic measures and long-term mortality in exercise test patients with clinically normal resting electrocardiograms. Am Heart J 158(1):61–70
Greenwell B, Boehmke B, Cunningham J, Developers G (2020) *gbm: Generalized Boosted Regression Models*. R package version 2.1.8. https://CRAN.R-project.org/package=gbm
Heath D, Kasif S, Salzberg S (1993) Induction of oblique decision trees, *in* 'Procedings of IJCAI-93', Vol. 1993, pp. 1002–1007
Hothorn T, Hornik K, Zeileis A (2006) Unbiased recursive partitioning: A conditional inference framework. J Comput Graph Stat 15(3):651–674
Irwin J (1949) The standard error of an estimate of expectation of life, with special reference to expectation of tumourless life in experiments with mice. Epidemiol Infect 47(2):188–189
Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS (2008) Random survival forests. Ann Appl Stat 2(3):841–860
Kim DH, Uno H, Wei L-J (2017) Restricted mean survival time as a measure to interpret clinical trial results. JAMA Cardiol 2(11):1179–1180
Klusowski JM, Tian PM (2024) Large scale prediction with decision trees. J Am Stat Assoc 119(545):525–537
Landwehr N, Hall MA, Frank E (2005) Logistic model trees. Mach Learn 59(1–2):161–205
Lee YD, Cook DH, Park J, Lee E-K (2013) Pptree: Projection pursuit classification tree. Electron J Stat 7(1):1369–1386
Liu Y, Xia Y (2022) 'ODRF: Consistency of the oblique decision tree and its random forest', arXiv preprint arXiv:2211.12653
Loh W-Y (2002) Regression trees with unbiased variable selection and interaction detection. Stat Sin 12:361–386
Loh W-Y (2014) Fifty years of classification and regression trees. Int Stat Rev 82(3):329–348
Loh W-Y, Shih Y-S (1997) Split selection methods for classification trees. Stat Sin 7(4):815–840
Lugosi G, Nobel AB (1996) Consistency of data-driven histogram methods for density estimation and classification. Ann Stat 24(2):687–706
Menze BH, Kelm BM, Splitthoff DN, Koethe U, Hamprecht FA (2011) On oblique random forests, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 453–469
Murthy SK, Kasif S, Salzberg S (1994) A system for induction of oblique decision trees. J Artif Intel Res 2:1–32
Nguyen P-H V, Yee R, Deshpande SK (2025) 'Oblique bayesian additive regression trees', *Transactions on Machine Learning Research* . Accepted by TMLR; published 16 Apr 2025. https://openreview.net/forum?id=l4Qnj4tHBx
Nobel AB (1996) Histogram regression estimation using data-dependent partitions. Ann Stat 24(3):1084–1105
Olson RS, La Cava W, Mustahsan Z, Varik G, Moore JH (2017) PMLB: A large benchmark suite for machine learning evaluation and comparison. BioData Mining 10(1):36
Palaniappan L, Lengerich BJ, Olson RS, Moore JH (2025) *pmlbr: R Interface to the Penn Machine Learning Benchmark (PMLB)*. R package version 0.3.0. https://CRAN.R-project.org/package=pmlbr
Quinlan JR (1992) Learning with continuous classes. 5th Australian Joint Conference on Artificial Intelligence 92:343–348
Rainforth T, Wood F (2015), 'Canonical correlation forests', arXiv preprint arXiv:1507.05444
Rodriguez JJ, Kuncheva LI, Alonso CJ (2006) Rotation forest: a new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell 28(10):1619–1630

Royston P, Parmar MK (2011) The use of restricted mean survival time to estimate the treatment effect in randomized clinical trials when the proportional hazards assumption is in doubt. Stat Med 30(19):2409–2421

Salzberg S (1991) A nearest hyperrectangle learning method. Mach Learn 6(3):251–276

Steele JM (1975) Combinatorial Entropy and Uniform Limit Laws, Ph.D. Thesis, Stanford University

Tibshirani J, Athey S, Sverdrup E, Wager S (2022) 'grf: Generalized random forests'. R package version 2.2.1. https://CRAN.R-project.org/package=grf

Tibshirani R (1996) Regression shrinkage and selection via the lasso. J Roy Stat Soc: Ser B (Methodol) 58(1):267–288

Tomita TM, Browne J, Shen C, Chung J, Patsolic JL, Falk B, Priebe CE, Yim J, Burns R, Maggioni M et al (2020) Sparse projection oblique randomer forests. J Mach Learn Res 21(1):4193–4231

Wettschereck D, Dietterich TG (1995) An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. Mach Learn 19(1):5–27

Zeileis A, Hothorn T, Hornik K (2008) Model-based recursive partitioning. J Comput Graph Stat 17(2):492–514

Zhan H, Liu Y, Xia Y (2025) 'Consistency of the oblique decision tree and its boosting and random forest', arXiv preprint arXiv:2211.12653 . Version 4, last revised 14 Feb 2025

Zhang T (2002) Covering number bounds of certain regularized linear function classes. J Mach Learn Res 2:527–550

Zharmagambetov A, Carreira-Perpiñán M (2020) Smaller, more accurate regression forests using tree alternating optimization, *in* H. Daumé III and A. Singh, eds, 'Proceedings of the 37th International Conference on Machine Learning', Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 11398–11408

Zhu H, Murali P, Phan D, Nguyen L, Kalagnanam J (2020) A scalable mip-based method for learning optimal multivariate decision trees. Adv Neural Inf Process Syst 33:1771–1781