# Minimax Rates and Spectral Distillation for Tree Ensembles

**Binh Duc Vu**
King's College London
binh.vu@kcl.ac.uk

**David S. Watson**
King's College London
david.watson@kcl.ac.uk

## Abstract

Tree ensembles such as random forests (RFs) and gradient boosting machines (GBMs) are among the most widely used supervised learners, yet their theoretical properties remain incompletely understood. We adopt a spectral perspective on these algorithms, with two main contributions. First, we establish minimax-optimal convergence for RF regression over a source class defined by a mixed partition kernel. The rate is governed by the kernel's eigenvalue decay and the partition complexity of its leading eigenspaces. This novel result sheds new light on the inductive bias of RFs. Second, we exploit this spectral viewpoint to develop compression schemes for tree ensembles. For RFs, leading eigenfunctions of the kernel operator capture the dominant predictive directions; for GBMs, leading singular vectors of the smoother matrix play an analogous role. Learning nonlinear maps for these spectral representations yields distilled models that are orders of magnitude smaller than the originals while maintaining competitive predictive performance. Our methods compare favorably to state of the art algorithms for forest pruning and rule extraction, with applications to resource constrained computing.

## 1 Introduction

Tree ensembles such as random forests (RFs) and gradient boosting machines (GBMs) are among the most popular and versatile supervised learning algorithms. Fast and easy to use, trees regularly attain state of the art results in tabular data tasks [45, 90], often beating out deep learning models that have proven more effective for image and language processing. Trees are especially good at adapting to sparse signals, making them a common choice in high-dimensional domains where parametric methods may struggle, such as bioinformatics [62], econometrics [18], and engineering [88].

Despite their prevalence, tree ensembles remain poorly understood at a theoretical level. For RFs, most authors focus on idealized variants such as purely random forests [20, 43, 10, 58], which make analysis more tractable but sacrifice the adaptive splitting that drives empirical performance. Finite sample generalization guarantees remain relatively rare. For GBMs, the situation is somewhat different. While boosting theory is relatively well developed [83, 26, 5, 108], the specific inductive bias of tree-based gradient boosting—and in particular, how ensemble size interacts with tree complexity—lacks a comparably sharp characterization.

We adopt a spectral perspective that addresses both algorithms. For RFs, we work in the kernel tradition [20, 31, 84], showing that the statistical rate is governed by the eigenvalue decay of the integral operator and how well spectral energy can be resolved via splits in the ambient feature space. Our main theoretical result is minimax optimality for an algorithm that closely approximates the original method of Breiman [21] over a source class of mixed partition kernels. The proof sheds light on how greedy splits inherently adapt to sparse signals.

Taking this spectral viewpoint, we observe that the principal eigenvectors of the RF kernel operator capture the most informative dimensions of the model. GBMs, by contrast, do not admit a kernel

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

representation, but an analogous compression scheme is achievable via the singular value decomposition of its implicit smoother matrix [30]. In both cases, models can be distilled by learning the nonlinear mapping from the input space to an appropriate spectral representation. Since a few leading modes capture most of the problem's variance, the resulting models maintain competitive performance while being significantly smaller and faster than the originals, thus enabling deployment on resource-constrained devices where full ensembles are infeasible. Crucially, while previous methods require users to set an accuracy tolerance—leaving the final model size unpredictable—our algorithm allows the size to be set explicitly. This provides a practical solution for deployment within TinyML settings with strict memory budgets [64], while remaining competitive on the empirical rate-distortion frontier.

The remainder of this paper is structured as follows. We review related work in Sect. 2. Following a brief summary of our setup and assumptions in Sect. 3, we state our main theoretical result in Sect. 4. This motivates a distillation scheme for tree ensembles, which we describe in Sect. 5. Empirical results are presented in Sect. 6, including extensive benchmarks against state of the art methods. A brief discussion follows in Sect. 7. Sect. 8 concludes.

## 2 Related Work

**RF Theory** For good overviews of RF theory, see [11, 85]. Early work in this area tended to prioritize asymptotic consistency results [22, 70, 12, 13, 43, 86], often under simplifying assumptions. Convergence rates have only gained attention more recently, with risk bounds established for RF classification [68, 41, 42] and individual regression trees [57, 89, 110, 105]. RF regression has typically been studied in the sparse additive regime, which is difficult for trees to resolve, though promising results have been derived under a local gain assumption [92, 29, 69]. Minimax convergence, a common optimality target in nonparametric regression, has proven challenging. Suboptimal rates have been derived for non-adaptive [58] and adaptive RFs [59]. An online variant of the RF algorithm known as Mondrian forests [60] are minimax optimal over a Hölder class [73, 74], as are tessellation forests [75, 76], which use oblique splits. However, the split mechanism in both cases is non-adaptive, and so these results do little to explain the performance of standard RFs. Globally optimal trees attain the minimax rate over a modified Besov class [105], but this has little bearing on greedy procedures. The only minimax result we are aware of for any adaptive RF variant relies on a modified split criterion and leaf-wise regression [109], two major deviations from Breiman's algorithm.

Our theoretical approach differs markedly from previous work in this area. Classical minimax results assume that the sampling distribution is dominated by some reference measure over the input space, giving rates that scale poorly with data dimensionality [91, 97]. However, in high-dimensional settings, data are often concentrated on a low-dimensional manifold [9]. Algorithms that can adapt to such sparse structures may converge far faster than theory would suggest. Building on this insight, several authors have taken a spectral approach to kernel regression, deriving novel minimax rates under a source condition and assumptions on eigenvalue decay [27, 15, 80, 107]. We adapt this approach to the RF context, establishing the minimax optimality of an adaptive RF algorithm.

**Ensemble compression** Pruning methods for tree ensembles selectively remove splits or entire trees from the forest. This tradition goes back to the original CART algorithm, where truncating depth was proposed as a form of tree regularization [23]. Numerous ensemble variants have been developed since [96], notably including Leaf Refinement with $L_1$ pruning (LRL1) [25], which aims to drop redundant trees; and ForestPrune (FP) [65], which optimizes tree depth on a per-tree basis. Though FP generally compresses better than LRL1, it produces symmetric trees by default, which may be suboptimal in some settings. Building on these methods, Devos et al. [33] propose Level-wise Optimization and Pruning (LOP), which traverses the ensemble by tree depth, optimizing a global $L_1$-regularized objective to simultaneously prune subtrees and update leaf predictions. These methods generally fix an error tolerance upfront and minimize model size subject to this budget.

Distillation approaches, by contrast, typically fix a target size upfront and minimize error subject to a compression rate budget. The method originated in deep learning, where knowledge distillation involves training a "student" network to mimic the soft labels of a larger "teacher" network [51, 40]. The approach has been adapted to tree ensembles. Zhou and Hooker [111] use the parent ensemble as an oracle to generate pseudo-samples and stabilize the decision boundaries of a single surrogate tree, while Vidal and Schiffer [99] develop a dynamic program to construct a minimal decision tree

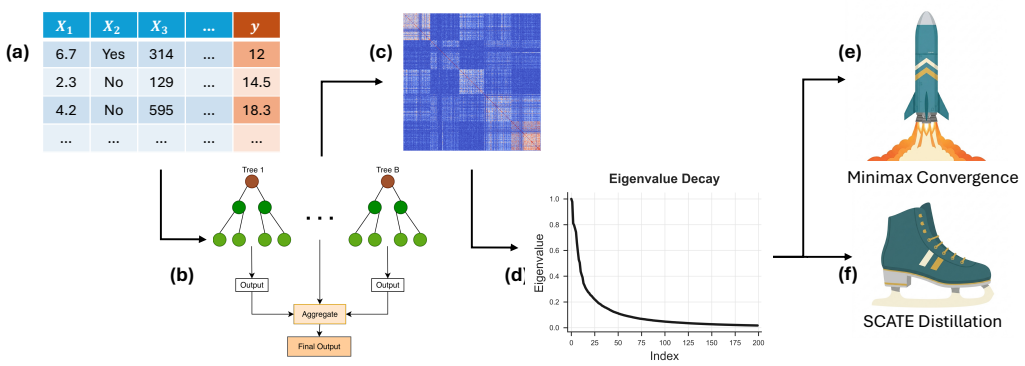



Figure 1: Summary of our pipeline. (a) Start with tabular training data for supervised learning (regression or binary classification). (b) Train a RF or GBM. (c) A kernel matrix $\mathbf{K}$ or smoother matrix $\mathbf{S}$ is extracted from the ensemble. (d) Compute the spectrum of the operator. (e) Under fast decay, RFs converge at the minimax-optimal rate. (f) The base ensemble is distilled by SCATE, which approximates model predictions by learning the top spectral directions.

that perfectly reproduces the original ensemble's predictions. Such exact distillation of a large forest into a single tree is NP-hard, and quickly becomes infeasible as dimensionality increases.

A related but distinct line of work aims to produce smaller, more interpretable models via rule extraction. Prominent early examples include RuleFit [39] and Node Harvest [71], which select the most informative splits via lasso penalties and constrained quadratic programming, respectively. More recent methods, such as SIRUS [7, 8] and FIRE [66, 67], generally produce stabler results. The SSF algorithm extracts nodes into isolated decision stumps [2], while MicroForest [106] and MILP-based alternatives [17] reassemble high information-gain nodes from multiple source forests into a new architecture. Although these methods enable fast, memory-efficient prediction on constrained devices, they are generally optimized for interpretability over performance, as we confirm in our experiments.

## 3 Background

Consider the standard regression setting, with data $\mathcal{D}_N = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ sampled i.i.d. from the joint probability measure $\mu$ over the feature space $\mathcal{X} \subseteq \mathbb{R}^d$ and label space $\mathcal{Y} \subseteq \mathbb{R}$. The goal is to estimate the conditional expectation $f^*(\boldsymbol{x}) = \mathbb{E}_\mu[Y \mid \boldsymbol{x}]$.

The CART algorithm [23] approximates $f^*$ by learning a recursive partition of $\mathcal{X}$. Axis-aligned splits are selected on a greedy basis to minimize a loss function over the training set $\mathcal{D}_N$, typically mean squared error (MSE) for continuous $Y$. Splits continue until either samples are exhausted or some stopping criterion is met, e.g. target depth. Terminal nodes are labeled by taking leaf-wise averages of $Y$. This procedure is universally consistent under common regularity assumptions [34, 47]. We distinguish between *oracle* CART, which always picks the optimal split point at a given node, and *empirical* CART, which may fail to do so due to sample noise. Note that oracle CART is still a greedy algorithm, and will therefore generally fail to converge on the globally optimal binary tree.

Despite their attractive asymptotic properties, trees can be high-variance predictors in practice. RFs overcome this by taking an ensemble of independently randomized CART basis functions, where each tree is trained on a bootstrap sample of the data and candidate splits are evaluated for only a subset of features. GBMs, on the other hand, are generally not randomized and grown to relatively shallow depth. Most importantly, they are trained sequentially, with each tree improving on the performance of the last. Both models are *smoothers* [49], i.e. supervised learners of the form

$$f(\boldsymbol{x}_0) = \sum_{i=1}^{N} w_i(\boldsymbol{x}_0)\, y_i, \tag{1}$$

where $\boldsymbol{x}_0 \in \mathcal{X}$ is an arbitrary point and $i \in [N]$ indexes the training data. Smoothers make predictions by taking linear combinations of training labels, with weights given by a potentially nonlinear smoothing function $w : \mathcal{X} \mapsto \mathbb{R}^N$. The differences between RFs and GBMs can be fully explained by their respective smoothing functions.

**Random Forests** In the case of RFs, the smoother is a kernel function. Each tree induces a partition of $\mathcal{X}$ into hyperrectangular cells $\{\mathcal{X}_\ell\}_{\ell=1}^M$, one for each leaf. Let $A(\cdot; b) : \mathcal{X} \mapsto \{\mathcal{X}_\ell\}_{\ell=1}^M$ be a map from samples to cells for a given tree $b$. The partition defines an indicator kernel that reports whether a pair of feature vectors are routed to the same leaf: $\hat{k}^{(b)}(\boldsymbol{x}, \boldsymbol{x}') = \mathbf{1}\big\{A(\boldsymbol{x}; b) = A(\boldsymbol{x}'; b)\big\}$. Normalizing over training data and averaging across trees, we derive the empirical RF kernel:

$$\hat{k}_N^B(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{B}\sum_{b=1}^{B}\Big(\frac{\hat{k}^{(b)}(\boldsymbol{x}, \boldsymbol{x}')}{\sum_{i=1}^N \hat{k}^{(b)}(\boldsymbol{x}, \boldsymbol{x}_i)}\Big).$$

This kernel provides the weights of Eq. 1 for RFs. The resulting matrix $\hat{\boldsymbol{K}} \in [0,1]^{N\times N}$ is PSD, symmetric, and doubly stochastic. Because splits are adaptively selected to minimize a loss function, the kernel is not independent of outcomes $Y$. Still, the resulting RKHS is asymptotically universal and characteristic [100]. The empirical kernel $\hat{\boldsymbol{K}}$ approximates the population operator $\boldsymbol{K}$, defined w.r.t. infinite data and trees. We write $\nu$ to summarize the randomness of the tree-generating mechanism.

**Gradient Boosting** GBMs construct predictions by sequentially fitting trees to pseudo-residuals under MSE loss. Let $\boldsymbol{f}_b \in \mathbb{R}^N$ denote the vector of model predictions on training data at step $b$. Beginning with $\boldsymbol{f}_0 = \mathbf{0}$, the $b$-th step fits a CART tree to the residual vector $\boldsymbol{\varepsilon}^{(b-1)} = \boldsymbol{y} - \boldsymbol{f}_{b-1}$, resulting in the normalized indicator kernel $\hat{r}^{(b)}(\boldsymbol{x}, \boldsymbol{x}')$. Evaluating this kernel across all training pairs gives the single-tree smoother matrix $\hat{\boldsymbol{R}}^{(b)} \in [0,1]^{N\times N}$. The model updates according to $\boldsymbol{f}_b \leftarrow \boldsymbol{f}_{b-1} + \eta\hat{\boldsymbol{R}}^{(b)}\boldsymbol{\varepsilon}^{(b-1)}$, with learning rate $\eta \in (0,1]$. Thus, beginning with $\hat{\boldsymbol{S}}^{(0)} = \mathbf{0}$, each round's prediction can be written as $\boldsymbol{f}_b = \hat{\boldsymbol{S}}^{(b)}\boldsymbol{y}$ with the accumulated linear operator given by $\hat{\boldsymbol{S}}^{(b)} \leftarrow \hat{\boldsymbol{S}}^{(b-1)} + \eta\hat{\boldsymbol{R}}^{(b)}(\boldsymbol{I} - \hat{\boldsymbol{S}}^{(b-1)})$. Solving this recursion yields the final GBM smoother:

$$\hat{\boldsymbol{S}} = \hat{\boldsymbol{S}}^{(B)} = \boldsymbol{I} - \prod_{b=1}^{B}(\boldsymbol{I} - \eta\hat{\boldsymbol{R}}^{(b)}).$$

The corresponding smoothing function $\hat{s}_N^B : \mathcal{X} \mapsto \mathbb{R}^N$ provides the weights of Eq. 1 for GBMs.

Unlike the RF operator $\boldsymbol{K}$, the GBM smoother $\boldsymbol{S}$ is generally neither symmetric nor PSD. This precludes a direct eigendecomposition of the kind used for RFs. However, the singular value decomposition (SVD) $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ provides a natural spectral representation: the leading right singular vectors of $\boldsymbol{S}$ identify the directions in label space that the ensemble amplifies most, while the corresponding left singular vectors describe the associated structure in prediction space. Equivalently, we could take the eigendecomposition of the Gram matrix $\boldsymbol{S}\boldsymbol{S}^\top$, though there are computational advantages to using SVD instead. We detail pseudocode to construct $\boldsymbol{S}$ and $\boldsymbol{K}$ in Appx. D

## 4 Minimax Convergence

In this section, we present a novel minimax convergence result for RF regression. To the best of our knowledge, this constitutes the first such result for a RF based on greedy CART splits. We start with a simple boundedness assumption.

(A1) *Bounds*. The feature space is compact on the unit hypercube $\mathcal{X} = [0,1]^d$, with marginal density bounded away from 0 and $\infty$, and outcomes on the unit interval $\mathcal{Y} = [0,1]$.

We make no assumption about the structural form of $Y \mid \boldsymbol{X}$, allowing for arbitrary heteroskedasticity.

Our next assumptions characterize the spectral source class. All $L^2$ norms and inner products from here on out are taken w.r.t. $\mu_X$ unless otherwise specified. Let $\boldsymbol{P}_T$ be the leaf averaging operator, $\boldsymbol{P}_T f := \mathbb{E}_\mu[f(\boldsymbol{x}) \mid \sigma_T]$, where $\sigma_T$ denotes the $\sigma$-algebra associated with tree $T$, a recursive partition of axis-aligned splits. Let $\Pi$ be a probability distribution over partitions. (We use $b \in [B]$ to index trees grown by the RF, and $T \sim \Pi$ to denote a generic partition of $\mathcal{X}$ with $|T|$ leaves.)

(A2) *Reference operator*. For fixed $\boldsymbol{K}_\Pi := \mathbb{E}_{T\sim\Pi}\big[\boldsymbol{P}_T\big]$, there exists some $\alpha \in (0,1]$, $\zeta \in (0,\infty)$, and $g \in L^2$ with $\|g\|_{L^2} \le \zeta$ such that $f^* = \boldsymbol{K}_\Pi^\alpha g$ and $f^*(\boldsymbol{x}) \in [0,1]$ $\mu_X$-a.e.

Though the basic form of the ellipsoid defined by (A2) is standard for inverse problems [27, 80], the definition of $\boldsymbol{K}_\Pi$ as a convex combination of partition projections is tailored to the RF setting. This

operator is positive, self-adjoint, and contractive. Next, we impose additional spectral structure. Let $(\psi_i, \lambda_i)_{i\geq 1}$ denote the nonconstant eigenpairs of $\boldsymbol{K}_\Pi$, ordered by decreasing eigenvalue.

(A3) *Polynomial decay*. The operator $\boldsymbol{K}_\Pi$ has eigenvalues $\lambda_i \asymp i^{-\beta}$, for some $\beta > 1$.
(A4) *Spectral localization*. For all $m \geq 1, \sup_{\omega\in\{-1,1\}^m} \left\| \sum_{j=m+1}^{2m} \omega_{j-m}\psi_j \right\|_\infty \lesssim m^{\alpha\beta+1/2}$.

(A3) specifies a spectral decay rate widely assumed in the analysis of kernel ridge regression [15, 107] and adaptive neural kernels [63, 14, 6]. (A4), which is only relevant for the minimax lower bound, ensures that bounded spectral perturbations remain valid regression functions. We write $\mathcal{F} := \mathcal{F}(\boldsymbol{K}_\Pi, \alpha, \zeta, \beta)$ to denote the class of functions defined by (A2)-(A4), and parameterize smoothness henceforth via $s := 2\alpha\beta$.

We make the following regularity assumptions on tree growth:

(A5) *Subsampling*. Trees are trained on subsamples of size $N^{(b)} = aN$ for some $a \in (0, 1]$.
(A6) *Budget*. Splits are selected on a best-first basis until a predetermined leaf budget is spent.
(A7) *Tiebreak*. If every candidate split has zero gain, the algorithm splits the largest-mass leaf at the midpoint of a coordinate drawn uniformly at random.

(A5)-(A7) represent slight deviations from Breiman [21]'s classic algorithm, which trains on bootstraps rather than subsamples and grows to full depth by default. The subsampling assumption is standard in the literature [72, 101, 3], while growing to a leaf budget is common in practice, and widely implemented in standard RF software [77, 104]. Zero-gain events are rare for generic signals, so (A7) is primarily a theoretical convention.

Our most substantive assumption links the quality of tree splits, optimized for MSE in feature space, to spectral gain. Let the operator $\boldsymbol{G}_m := \sum_{i\leq m} \psi_i \otimes \psi_i$ project onto the leading $m$ eigenfunctions of $\boldsymbol{K}_\Pi$. Then spectral tree complexity can be quantified by:

$$\mathfrak{C}_m(\varepsilon) := \sup_{\boldsymbol{u}\in\mathrm{ran}(\boldsymbol{G}_m), \|\boldsymbol{u}\|_2=1} \inf\left\{|T| : \|(\boldsymbol{I} - \boldsymbol{P}_T)\boldsymbol{u}\|_{L_2}^2 \leq \varepsilon\right\}.$$

This value represents the number of leaves that an optimal tree requires in the worst case to resolve spectral energy of order $m$ up to tolerance $\varepsilon$. Let $V(T; f) := \|(\boldsymbol{I} - \boldsymbol{P}_T)f\|_{L_2}^2$ and $U_m(T; f) := \|(\boldsymbol{I} - \boldsymbol{P}_T)\boldsymbol{G}_m f\|_{L_2}^2$ represent unresolved energy of tree $T$ for function $f \in \mathcal{F}$ at full and $m$-truncated resolutions, respectively.

(A8) *Amortized spectral gain*. Fix $\varepsilon \in (0, 1)$. There exist constants $c \in (0, 1)$ and $C > 0$ such that, on any linear reachability window $|T| \lesssim m$, for every $m \geq 1$ and $f \in \mathcal{F}$, some $h \lesssim \mathfrak{C}_m(\varepsilon)$ further CART splits satisfy

$$V(T; f) - \mathbb{E}_\nu\big[V(T_{+h}; f) \mid T\big] \geq c\, U_m(T; f) - C\, \|(\boldsymbol{I} - \boldsymbol{G}_m)f\|_{L^2}^2. \tag{ASG}$$

(A8) says that, within $\mathfrak{C}_m(\varepsilon)$ splits, the forest captures a constant fraction of whatever leading $m$-dimensional spectral energy remains, up to the irreducible tail. The condition may appear somewhat ad hoc at first glance, but ASG is in fact weaker than the structural impurity decrease assumption, which has recently become popular in RF theory [29, 92, 57, 69]; see Appx. B.5 for a comparison.

With these elements in place, we state our main result. The proof is deferred to Appx. A.

**Theorem 4.1** (Minimax optimality)**.** *Assume (A1)-(A8), with $s := 2\alpha\beta$ and $\mathfrak{C}_m(\varepsilon) = \mathcal{O}(m)$ for some $\varepsilon \in (0, 1)$. Then a RF $\hat{f}_{\mathrm{RF}}$ with oracle CART splits, $B = \infty$ trees, and leaf budget $M \asymp N^{1/(s+1)}$ satisfies*

$$\sup_{f^*\in\mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \lesssim N^{-\frac{s}{s+1}}.$$

*This rate is minimax optimal over the source class $\mathcal{F}$. Replacing oracle splits with empirical splits preserves the rate up to a logarithmic factor $\log(Nd)^{\frac{s}{s+1}}$ under an empirical variant of ASG with $M \asymp \big(N/\log(Nd)\big)^{1/(s+1)}$ leaves.*

For a precise statement of our empirical ASG condition, see Appx. A.3. The infinite-tree idealization is benign, and can be removed at the cost of an additive $1/B$ term, a negligible offset for $B$ on the order of a few hundred. The more substantive condition is that ASG applies with $\mathfrak{C}_m = \mathcal{O}(m)$. This

assumption is violated by some well-studied function classes, including the sparse additive model that has been the focus of much RF theory for over a decade [86, 92, 29, 59]. This accounts for the negative results of Shuo Tan et al. [89], who show that even an optimal tree attains suboptimal rates in such settings (though their own experiments suggest faster rates for RFs).

However, there is a large class of natural multivariate regression models that fall within our minimax regime. In Appx. B, we study a hierarchically separated tree-Besov class, in which the target function may exhibit different sparse and anisotropic structure across cells of a recursive partition. This class is minimax-hard, with exponentially many independently varying fine-scale constrasts, but better suited to the inductive bias of RFs than additive models. Our construction can be viewed as a greedy-accessible analogue of the PSHAB spaces studied by Xu et al. [105], whose recursive-tree approximation theory likewise falls in the linear-complexity regime. For a detailed discussion of related works and differing rates derived under varying assumptions, see Appx. C.

Thm. 4.1 provides a conceptual bridge to spectral distillation. Under (A3), the regression signal is concentrated in the leading eigenspaces of a reference operator with rapidly decaying spectrum, while (A8) posits that greedy RFs can efficiently resolve this spectral geometry. This immediately suggests a compression scheme: if the induced kernel inherits the spectral profile of $\boldsymbol{K}_\Pi$, then $\lambda_i(\hat{\boldsymbol{K}})$ decays rapidly and a low-rank approximation retaining only its leading eigenfunctions should preserve most of the predictive signal. Our present theory does not explicitly establish such convergence, so we treat this as a theoretically motivated hypothesis rather than a consequence of Thm. 4.1. The distillation procedure developed next tests the hypothesis directly, using the empirical spectrum of the fitted RF to identify and retain its dominant predictive modes.

## 5 Spectral Compression of Adaptive Tree Ensembles (SCATE)

We introduce **SCATE** (**S**pectral **C**ompression of **A**daptive **T**ree **E**nsembles). SCATE uses a (small) neural network to learn the leading spectral directions of an implicit linear operator—the kernel matrix $\hat{\boldsymbol{K}}$ for RFs, or the smoother matrix $\hat{\boldsymbol{S}}$ for GBMs. For an RF $\hat{f}_{\mathrm{RF}}$ trained on $N$ samples, let $(\boldsymbol{\Psi}, \boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda}))$ be the eigenpairs of the kernel matrix.[1] Let $\psi : \mathcal{X} \mapsto \mathbb{R}^N$ be a map from input features to $\hat{\boldsymbol{K}}$'s bundled eigenfunctions, treated as a column vector. Model predictions over the training data can be written $\hat{f}_{\mathrm{RF}}(\boldsymbol{X}) = \hat{\boldsymbol{K}}\boldsymbol{y}$, where $\boldsymbol{y}$ is a vector of training labels. For a test point $\boldsymbol{x}_0$, predictions rely on localized kernel weights $\hat{\boldsymbol{k}}_0 = \boldsymbol{\Psi}\boldsymbol{\Lambda}\psi(\boldsymbol{x}_0)$, yielding:

$$\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_0) = \hat{\boldsymbol{k}}_0^\top \boldsymbol{y} = (\boldsymbol{\Psi}\boldsymbol{\Lambda}\psi(\boldsymbol{x}_0))^\top \boldsymbol{y} = \psi(\boldsymbol{x}_0)^\top \boldsymbol{\Lambda}\boldsymbol{\Psi}^\top \boldsymbol{y}.$$

This derivation reveals our compression strategy: we store the eigendecomposition of $\hat{\boldsymbol{K}}$, truncated to rank $m \ll N$, and train a neural network to approximate the top $m$ eigenfunctions $\{\psi_i\}_{i=1}^m$. By defining a fixed coefficient vector $\boldsymbol{c} := \boldsymbol{\Lambda}_m \boldsymbol{\Psi}_m^\top \boldsymbol{y} \in \mathbb{R}^m$, test time predictions amount to evaluating $\hat{f}_{\mathrm{RF}}(\boldsymbol{x}_0) \approx \sum_{i=1}^m c_i\, \psi_i(\boldsymbol{x}_0)$. By the Eckart-Young-Mirsky theorem [37], this rank-$m$ truncation provides the optimal low-rank matrix approximation under the Frobenius norm, ensuring we retain maximal information from the ensemble's smoother matrix. An identical strategy works for GBMs, substituting $\hat{\boldsymbol{S}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ for $\hat{\boldsymbol{K}} = \boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Psi}^\top$ and learning the left singular vector function $u : \mathcal{X} \mapsto \mathbb{R}^m$. Note that, because we can compress any ensemble that generates a valid smoother matrix in accordance with Eq. 1, SCATE can be applied to a model already pruned by a different algorithm, with an exact smoother when the leaf values are not altered by the process (e.g., FP) , and approximate otherwise (e.g., LOP).

**Learning spectral directions with SCATE** Extracting eigenfunction mappings from a given matrix is a well-studied problem. Existing neural approaches, such as SpIN [78] and NeuralEF [32], rely heavily on constrained optimization, explicitly enforcing eigenvector orthonormality within the loss function to guide the network. However, constrained optimization methods are computationally heavy, making them difficult to tune and design in low-size settings. Alternatively, methods like NeuralSVD [82] utilize unconstrained optimization by assigning greater weight to eigenfunctions with stronger signals. However, in practice, we find that these methods struggle to learn efficiently in constrained settings due to their complex objectives—see Appx. E.1 for a small comparison.

[1] Though we generally omit hats on eigenvectors and eigenvalues for notational convenience, it is worth clarifying that $(\boldsymbol{\Psi}, \boldsymbol{\Lambda})$ represent finite sample estimates, not population parameters. The same goes for the SVD of $\hat{\boldsymbol{S}}$.

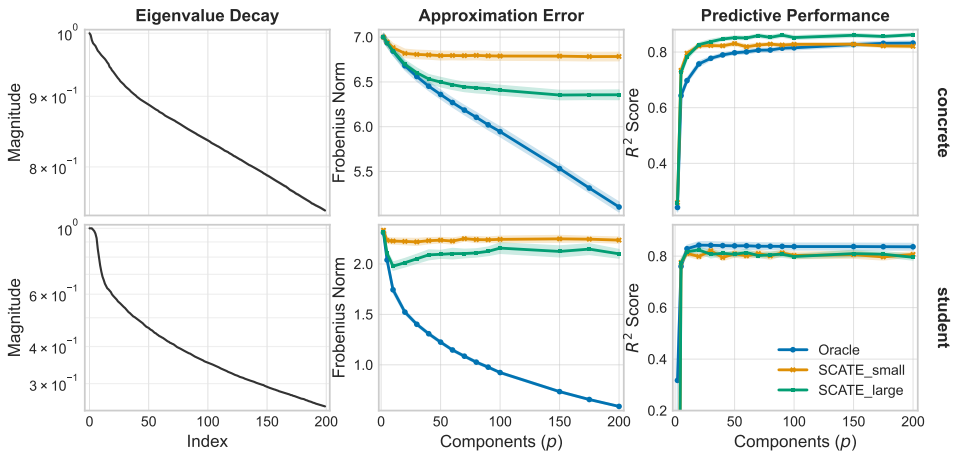



Figure 2: Spectral decay, kernel approximation error, and predictive performance across two datasets.

To overcome these limitations, SCATE focuses on a simple architecture with a targeted loss formulation. The network is parameterized as a simple Multi-Layer Perceptron (MLP) with SiLU activations [38]. SCATE is trained via a multi-task objective [28], which minimizes a spectrum-weighted MSE alongside an orthogonality penalty, $\mathcal{L}_{\text{ortho}} = m^{-2} \| \operatorname{offdiag}(b^{-1} \widehat{\Psi}^\top \widehat{\Psi}) \|_F^2$, where $\widehat{\Psi} \in \mathbb{R}^{b \times m}$ contains the network's $m$ outputs on a minibatch of $b$ points.

$$\min_g \frac{1}{b} \sum_{i \in \mathcal{B}} \sum_{j=1}^{m} \frac{\lambda_j}{\lambda_1} \big( g_j(\boldsymbol{x}_i) - \psi_j(\boldsymbol{x}_i) \big)^2 + \gamma \mathcal{L}_{\text{ortho}}.$$

We explicitly weigh the MSE by the corresponding squared eigenvalue $\lambda_j^2$—or singular value $\sigma_j^2$ for SVD—to encode greater interest in the most informative, high-variance spectral directions, and set $\gamma = 0.001$, which we found balanced mode diversity and reconstruction error well.

**Oracle Benchmark** To isolate our network's approximation error from the inherent loss of a low-rank bottleneck, we benchmark against an eigenfunction oracle. Given direct access to the train-test cross-kernel matrix $\boldsymbol{K}_{test}$, the oracle computes the optimal least-squares mapping $\boldsymbol{W}$ from the scaled training eigenvectors $\boldsymbol{C} = \boldsymbol{\Psi}\boldsymbol{\Lambda}$, solving $\boldsymbol{W}\boldsymbol{C}^\top = \boldsymbol{K}_{test}$. Truncating to the top $m$ dimensions, the reconstructed test kernel is $\tilde{\boldsymbol{K}} = \boldsymbol{W}_m \boldsymbol{C}_m^\top$. The oracle then computes predictions via $\tilde{\boldsymbol{K}}\boldsymbol{y}$, and establishes a strict lower bound on the Frobenius reconstruction error $\|\boldsymbol{K}_{test} - \tilde{\boldsymbol{K}}\|_F$ achievable by any rank-$m$ distillation on the same spectral subspace. Interestingly, this does not guarantee that $\tilde{\boldsymbol{K}}$ attains optimal performance in terms of MSE on the associated prediction task, although it is rarely far off in practice.

## 6 Experiments

In this section, we evaluate SCATE's performance for both RFs and GBMs. Comprehensive details describing the experimental setup can be found in Appx. D; code is available online.[2] We use consistent hyperparameters throughout, with `num_trees=250`, `max_depth=15` for RFs and `num_trees=100`, `max_depth=6` for GBMs.

**Spectra and Generalization** Our strategy is motivated by the idea that the strong performance of RFs (and, by analogy, GBMs) is driven by their capacity to learn rich, informative embeddings encoded in the principal directions of their respective smoothing functions. Spectral decay has implications not just for the convergence rate of an ensemble, as in Thm. 4.1, but for distillation as well. To demonstrate this, Fig. 2 illustrates the predictive performance of SCATE and the spectral oracle as a function of the number of eigenfunctions ($m$) for the RF case. We highlight the `student` and `concrete` datasets as representative examples of "easy" (rapid eigendecay) and "hard" (slower eigendecay) scenarios, respectively. Results for additional datasets are provided in the Appx E.2.

[2] `https://github.com/binhducvu/SCATE`

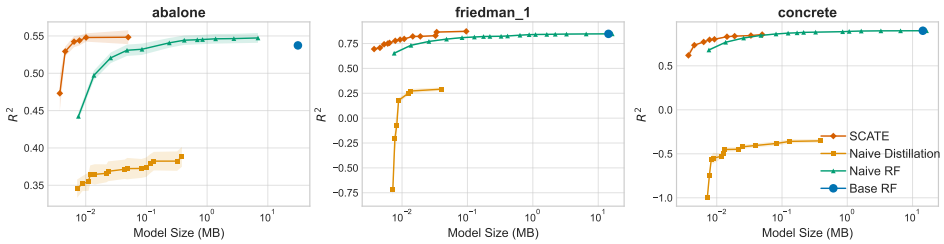



Figure 3: Rate-performance Pareto curves across three benchmark datasets. Shaded regions indicate standard errors across 10 trials.

We evaluate two different architectures for comparison—a small MLP of `width=16,depth=2` and a larger MLP of `width=128,depth=4`. While the larger network's superior performance across both datasets confirms that increased expressivity is useful, the results also demonstrate a connection between compressibility and difficulty, where the model performance converges quicker for the dataset with faster decay. Because SCATE's resource-bound architecture is limited to learning top components, its success depends on how much of the model's total predictive power is concentrated within these leading eigenfunctions.

**Naïve Benchmark** To further validate our approach, we benchmark SCATE against three naïve tree compression baselines: (1) a vanilla MLP mirroring SCATE's architecture, but trained to directly mimic the individual tree predictions of the RF (Naïve Distillation); (2) an RF artificially constrained to various sizes via hyperparemeter tuning (Naïve RF); and (3) the base performance of the original RF. Whereas in Fig. 2, we fixed model size and varied the number of eigenfunctions $m$, here we do the inverse. With fixed $m = 50$—a value we found to be generally robust, striking a reasonable balance between complexity (of the base ensemble) and learnability (for the compressed network)—we vary the model size to create a Pareto frontier of rate-distortion curves for various methods and datasets. The base RF provides a reference point, indicating an optimal performance threshold (albeit at high model size).

We visualize results for three datasets in Fig. 3 (additional results provided in Appx. E.3). SCATE consistently dominates in this experiment, effectively matching the predictive performance of the base RF while using orders of magnitude less memory. Naïve RF approaches the same performance levels but at a far slower rate. SCATE also significantly outperforms the Naïve Distillation scheme, which struggles to approximate RF predictions using the comparatively impoverished signal provided by direct RF outputs. This aligns with a notable finding of Hinton et al. [51], who initiated research into neural network distillation. They showed that small student networks can quickly attain strong performance in classification tasks when trained on softmax weights from a larger teacher network because these so-called "soft labels" provide far more information than the hard labels used to train the original model. We find that eigenfunctions perform a similar role in forest distillation for regression, accelerating distillation with rich spectral signals.

**Compression Benchmark** For this experiment, we evaluate the size-performance trade-off of SCATE against a suite of established compression techniques, including pruning (FP and LOP), rule extraction (TreeExtract and FIRE), and approximate distillation (SSF). We also introduce a hybrid approach, FP-SCATE, which applies ForestPrune to a base model before further refinement with SCATE. We excluded other techniques like SIRUS, RuleFit, and LRL1, as prior studies suggest they are dominated by the methods we use here. We were unable to find an implementation of MicroForest. For all SCATE variants, we maintain $m = 50$ eigenfunctions, consistent with our previous analysis.

We evaluate performance across several regression and classification datasets using $R^2$ and accuracy, respectively; further dataset details are provided in Appx. D. Because the benchmarked algorithms utilize different hyperparameter mechanisms to control compression rates, we adopt a size-constrained evaluation strategy. For each method and run, we sweep across a broad hyperparameter range to generate a performance-vs-size profile. We then report the best performance achieved at two fixed model sizes: 10KB and 100KB. These thresholds reflect the limited flash memory typical of microcontrollers, which generally operate with a budget under 1MB [19]. Original model sizes vary, but at 10KB, compression factors are on the order of $10^3$ for RFs and $10^2$ for GBMs — see Appx. E.4. This experimental design is applied to both RF and GBM models, with the exception of SSF, which

Table 1: RF compression benchmark, averaged over up to 10 trials. Best results per row and results within one standard error of the best are in bold, with values being $R^2$ for regression (top half) and accuracy for classification (bottom half). NA indicates no within-budget candidate; TO indicates no candidate result within time limit. For an expanded table with standard errors and run counts, see Table 6.

| | | Size = 10KB | | | | | | | Size = 100KB | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **RF** | **FP** | **LOP** | **FIRE** | **TE** | **SSF** | **SCATE** | **FP-SCATE** | **FP** | **LOP** | **FIRE** | **TE** | **SSF** | **SCATE** | **FP-SCATE** |
| abalone | 0.55 | 0.35 | 0.50 | 0.28 | NA | NA | **0.56** | 0.54 | 0.53 | 0.51 | 0.28 | 0.55 | NA | **0.57** | 0.55 |
| auto_mpg | 0.87 | 0.63 | 0.81 | 0.68 | NA | NA | 0.85 | **0.87** | 0.82 | 0.82 | 0.68 | 0.84 | 0.81 | 0.87 | **0.88** |
| bike | 0.94 | 0.43 | NA | TO | TO | **0.70** | 0.68 | 0.68 | 0.72 | NA | TO | TO | 0.70 | **0.86** | 0.84 |
| boston | 0.85 | 0.60 | **0.80** | 0.68 | NA | NA | 0.78 | **0.79** | 0.80 | 0.82 | 0.69 | 0.81 | 0.72 | 0.81 | **0.83** |
| california | 0.80 | 0.45 | 0.68 | TO | TO | NA | 0.70 | **0.71** | 0.68 | **0.74** | TO | TO | NA | **0.75** | **0.74** |
| concrete | 0.91 | 0.50 | 0.74 | 0.73 | NA | NA | 0.80 | **0.82** | 0.76 | 0.85 | 0.75 | 0.89 | **0.90** | 0.86 | 0.86 |
| diamonds | 0.98 | 0.78 | NA | TO | TO | NA | **0.94** | 0.59 | 0.93 | NA | TO | TO | 0.93 | **0.96** | 0.60 |
| friedman_1 | 0.85 | 0.47 | 0.76 | 0.48 | NA | NA | **0.82** | 0.81 | 0.74 | 0.82 | 0.49 | 0.79 | NA | **0.86** | 0.86 |
| student | 0.85 | 0.64 | 0.80 | 0.71 | NA | 0.82 | **0.85** | 0.84 | 0.84 | 0.81 | 0.72 | 0.79 | 0.82 | **0.86** | 0.84 |
| wq | 0.50 | 0.24 | 0.31 | 0.31 | NA | NA | **0.34** | **0.33** | 0.35 | **0.36** | 0.34 | 0.28 | 0.24 | **0.36** | **0.36** |
| adult | 0.85 | 0.79 | 0.83 | TO | TO | NA | **0.84** | 0.83 | 0.84 | 0.83 | TO | TO | **0.85** | **0.85** | 0.85 |
| banknote | 0.99 | 0.87 | 0.99 | 0.99 | NA | 1.00 | **1.00** | **1.00** | 0.92 | 0.99 | 0.99 | 0.89 | **1.00** | **1.00** | **1.00** |
| bc | 0.96 | 0.95 | 0.92 | 0.96 | NA | 0.98 | **0.98** | 0.98 | 0.95 | 0.92 | 0.96 | 0.92 | 0.98 | **0.99** | **0.99** |
| churn | 0.86 | 0.82 | 0.84 | 0.85 | NA | NA | **0.86** | 0.85 | 0.85 | 0.85 | 0.85 | 0.85 | 0.83 | **0.86** | 0.86 |
| diabetes | 0.76 | 0.76 | 0.74 | 0.77 | NA | NA | **0.79** | **0.79** | 0.76 | 0.74 | 0.77 | 0.66 | 0.73 | **0.79** | **0.80** |
| magic | 0.88 | 0.79 | 0.82 | **0.86** | NA | NA | **0.86** | **0.86** | 0.85 | 0.86 | 0.87 | 0.84 | NA | **0.87** | 0.87 |
| spambase | 0.95 | 0.87 | 0.92 | **0.94** | NA | NA | 0.93 | 0.93 | 0.91 | 0.93 | **0.94** | 0.89 | **0.94** | 0.94 | 0.94 |
| telco | 0.79 | 0.75 | 0.77 | 0.79 | NA | NA | **0.80** | 0.80 | 0.80 | 0.77 | 0.79 | 0.79 | 0.78 | **0.81** | 0.80 |
| **Average Rank** | | 4.67 | 3.86 | 4.08 | 6.44 | 5.50 | **1.53** | 1.92 | 4.22 | 4.72 | 5.31 | 5.19 | 4.67 | **1.56** | 2.33 |

is restricted to RFs. Notably, while the RF kernel $\boldsymbol{K}$ generalizes naturally to both tasks, the GBM smoother $\boldsymbol{S}$ is defined specifically for MSE loss. To extend SCATE to GBM classification, we train the base model using MSE loss—effectively treating classification as a regression problem—which limits our evaluation to the binary case. As shown in Tables 1 and 2, SCATE yields highly competitive results for both RF and GBM compression, securing the highest average rank across all evaluated model sizes, with FP-SCATE coming in second for the RF benchmark. In contrast, this hybrid method declines markedly when applied to GBMs. We attribute this decline to the weaker baseline performance of FP as a preprocessing step, compounded by the fact that FP's pruning mechanism alters the ensemble structure in a way that violates our recursive formula for computing $\boldsymbol{S}$. For a more comprehensive results of algorithm performance, see Appx. E.

## 7 Discussion

Our theoretical results suggest that generalization in tree ensembles is strongly linked to the compressibility of the model. The more information is encoded in the leading eigenfunctions of the RF or singular vectors of the GBM, the better the model learns, and the more easily it can be distilled. This accords with the minimum description length principle [81, 4, 52, 46] and related information theoretic insights [95, 50] suggesting a profound connection between the dual tasks of prediction and compression.

Notably, our minimax optimality theorem is limited to RF regression. A promising direction for future work would be to pursue similar results for GBMs. This poses substantial technical challenges, as kernel methods do not apply. The smoother matrix is neither symmetric nor PSD, and boosting induces nontrivial dependencies between basis functions. Still, recent work suggests some theoretical overlap in the mechanics of tree ensembles that may be exploited for future results [112].

Our compression method differs fundamentally from traditional techniques that learn directly from the base model. Instead, SCATE reproduces the leading spectral directions of a linear operator. This approach offers distinct advantages over classical pruning methods, which typically optimize the tree structure itself, making their computational cost heavily dependent on the base model's size. By contrast, SCATE relies solely on the kernel, which can be computed with a single forward pass of the training data. Scaling up the complexity of the base model therefore has lower impact on distillation time. Furthermore, SCATE compiles into a small network where inference amounts to computing fast matrix operations, maintaining the parallelization capabilities of its teacher model.

Table 2: GBM compression benchmark, averaged over up to 10 trials. Best results per row and results within one standard error of the best are in bold, with values being $R^2$ for regression (top half) and accuracy for classification (bottom half). NA indicates no within-budget candidate; TO indicates no candidate result within time limit. For an expanded table with standard errors and run counts, see Table 7.

| | | Size = 10KB | | | | | | Size = 100KB | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **GBM** | **FP** | **LOP** | **TE** | **FIRE** | **SCATE** | **FP-SCATE** | **FP** | **LOP** | **TE** | **FIRE** | **SCATE** | **FP-SCATE** |
| abalone | 0.53 | 0.44 | 0.50 | 0.54 | 0.46 | **0.56** | 0.55 | 0.53 | 0.52 | 0.55 | 0.46 | **0.57** | 0.55 |
| auto_mpg | 0.86 | 0.80 | 0.82 | **0.85** | 0.82 | **0.85** | 0.83 | 0.86 | 0.83 | 0.85 | 0.82 | **0.87** | 0.85 |
| bike | 0.95 | 0.57 | NA | TO | **0.88** | 0.62 | 0.62 | 0.92 | **0.94** | TO | 0.88 | 0.92 | 0.87 |
| boston | 0.86 | 0.78 | **0.81** | **0.79** | 0.78 | **0.79** | 0.71 | **0.87** | 0.83 | 0.82 | 0.78 | 0.84 | 0.76 |
| california | 0.83 | 0.58 | 0.76 | **0.79** | 0.75 | 0.71 | 0.70 | 0.80 | **0.82** | 0.79 | 0.75 | 0.79 | 0.76 |
| concrete | 0.92 | 0.71 | 0.83 | **0.87** | 0.84 | 0.79 | 0.81 | 0.90 | 0.91 | **0.92** | 0.84 | 0.88 | 0.87 |
| diamonds | 0.98 | 0.88 | TO | NA | **0.94** | **0.94** | 0.93 | 0.97 | TO | **0.98** | 0.94 | 0.97 | 0.96 |
| friedman_1 | 0.91 | 0.68 | 0.87 | **0.92** | 0.79 | 0.89 | 0.87 | 0.90 | 0.90 | 0.92 | 0.79 | **0.97** | 0.93 |
| student | 0.83 | 0.81 | 0.79 | 0.79 | 0.75 | **0.84** | 0.70 | **0.86** | 0.79 | 0.79 | 0.75 | 0.84 | 0.70 |
| wq | 0.45 | 0.29 | 0.32 | 0.30 | **0.35** | 0.33 | 0.33 | **0.39** | **0.38** | 0.30 | 0.35 | 0.36 | 0.36 |
| adult | 0.85 | 0.82 | 0.84 | **0.85** | 0.81 | 0.85 | 0.84 | 0.85 | 0.85 | **0.86** | 0.81 | 0.85 | 0.85 |
| banknote | 0.98 | 0.95 | 0.98 | 0.99 | 0.98 | **1.00** | 1.00 | 0.99 | 0.98 | 1.00 | 0.98 | **1.00** | 1.00 |
| bc | 0.94 | 0.94 | 0.94 | 0.94 | 0.95 | **0.98** | 0.95 | 0.95 | 0.94 | 0.95 | 0.95 | **0.99** | 0.95 |
| churn | 0.86 | 0.85 | 0.86 | **0.86** | 0.85 | 0.86 | 0.85 | **0.86** | 0.86 | **0.87** | 0.85 | **0.86** | 0.86 |
| diabetes | 0.74 | 0.77 | 0.76 | 0.78 | 0.74 | **0.79** | 0.73 | 0.78 | 0.76 | 0.78 | 0.74 | **0.80** | 0.73 |
| magic | 0.88 | 0.81 | 0.85 | **0.87** | 0.85 | 0.86 | 0.86 | 0.87 | 0.87 | **0.88** | 0.85 | 0.87 | 0.87 |
| spambase | 0.95 | 0.90 | 0.92 | **0.94** | 0.92 | 0.93 | 0.93 | 0.95 | 0.93 | **0.95** | 0.92 | 0.94 | 0.94 |
| telco | 0.79 | 0.79 | 0.79 | **0.80** | 0.79 | **0.81** | 0.79 | 0.80 | 0.79 | 0.81 | 0.79 | **0.81** | 0.80 |
| **Average Rank** | | 5.28 | 3.67 | 2.50 | 3.83 | **2.06** | 3.67 | 2.64 | 3.94 | 2.67 | 5.50 | **2.06** | 4.19 |

SCATE is not without its limitations. First, training imposes nontrivial computational demands: storing operators requires $\mathcal{O}(N^2)$ space, and truncated decomposition takes $\mathcal{O}(N^2 m)$ time. However, alongside standard mitigations like randomized SVD [48] and Nyström approximations [103], we find empirically that subsampling the training set to 10,000 observations produces no noticeable loss in distillation accuracy, effectively removing this bottleneck for large datasets. Second, deployment constraints restrict SCATE to using just the top 20 to 50 eigenfunctions. This may be insufficient for complex problems, leading SCATE's approximation quality to suffer—although our strong empirical results indicate this is not often the case in practice. While our framework allows us to stipulate the final size of the model (see Appx. D.4 for a basic derivation of model size given architecture), choosing the optimal network architecture is nontrivial. In our experiments, a simple MLP proved effective in the resource-constrained setting, but more sophisticated alternatives may provide improvements. Finally, our distilled model sacrifices the purported interpretability benefits of trees, which naturally lend themselves to rule extraction and formal verification [55, 54].

Finally, we remark that the RF and GBM operators share a conceptual symmetry to two adaptive kernel interpretations of neural networks: the conjugate kernel (CK) [61] and the neural tangent kernel (NTK) [56]. In particular, the RF kernel and the CK both construct a symmetric, static matrix by averaging over random projections. Conversely, the GBM smoother and NTK both instead track the learning trajectory of residual fitting during optimization. Adapting our results to deep learning architectures is a natural direction for future work.

## 8 Conclusion

We have shown that tree ensembles can be compressed into significantly smaller models of comparable performance using spectral distillation. These results are motivated by a novel theoretical insight: that RFs are minimax-optimal w.r.t. a source class of mixed partition kernels when eigenvalues decay at a polynomial rate. To our knowledge, this is the first minimax result for an RF algorithm composed of ordinary CART basis functions. Our key assumption is amortized spectral gain, a weaker variant of the popular structural impurity decrease assumption. We conjecture that a similar result holds for GBMs, when singular values of the smoother matrix decay quickly. Extensive benchmarks show that our methods are competitive with state of the art pruning and rule extraction techniques for ensemble compression, providing new methods for resource constrained computing.

## Acknowledgements

We thank Erwan Scornet and several anonymous reviewers for their helpful comments on an earlier draft of this manuscript. This research was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) [Grant reference number EP/Y035216/1] Centre for Doctoral Training in Data-Driven Health (DRIVE-Health) at King's College London.

## References

[1] Nathalie Akakpo. Adaptation to anisotropy and inhomogeneity via dyadic piecewise polynomial selection. *Mathematical Methods of Statistics*, 21(1):1–28, 2012.

[2] Fouad Alkhoury and Pascal Welke. Splitting stump forests: tree ensemble compression for edge devices. In *International Conference on Discovery Science*, pages 3–18. Springer, 2024.

[3] Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *Ann. Statist.*, 47(2): 1148–1178, 2019.

[4] Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

[5] Peter L. Bartlett, Peter J. Bickel, Peter Bühlmann, Yoav Freund, Jerome Friedman, Trevor Hastie, Wenxin Jiang, Michael J. Jordan, Vladimir Koltchinskii, Gábor Lugosi, Jon D. McAuliffe, Ya'acov Ritov, Saharan Rosset, Robert E. Schapire, Robert Tibshirani, Nicolas Vayatis, Bin Yu, Tong Zhang, and Ji Zhu. Discussions of boosting papers, and rejoinders. *The Annals of Statistics*, 32(1):85 – 134, 2004.

[6] Yuval Belfer, Amnon Geifman, Meirav Galun, and Ronen Basri. Spectral analysis of the neural tangent kernel for deep residual networks. *Journal of Machine Learning Research*, 25(184):1–49, 2024.

[7] Clément Bénard, Gérard Biau, Sébastien da Veiga, and Erwan Scornet. Interpretable random forests via rule extraction. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 937–945, 2021.

[8] Clément Bénard, Gérard Biau, Sébastien Da Veiga, and Erwan Scornet. SIRUS: Stable and Interpretable RUle Set for classification. *Electronic Journal of Statistics*, 15(1):427 – 505, 2021.

[9] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[10] Gérard Biau. Analysis of a random forests model. *J. Mach. Learn. Res.*, 13:1063–1095, 2012.

[11] Gérard Biau and Erwan Scornet. A random forest guided tour. *TEST*, 25(2):197–227, 2016.

[12] Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *J. Mach. Learn. Res.*, 9(66):2015–2033, 2008.

[13] Gérard Biau and Luc Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *J. Multivar. Anal.*, 101(10): 2499–2518, 2010.

[14] Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] Gilles Blanchard and Nicole Mücke. Kernel regression, minimax rates and effective dimensionality: Beyond the regular case. *Analysis and Applications*, 18(4):683–696, 2020.

[16] Gilles Blanchard, Christin Schäfer, Yves Rozenholc, and Klaus-Robert Müller. Optimal dyadic decision trees. *Machine Learning*, 66(2–3):209–241, 2007.

[17] Lorenzo Bonasera and Emilio Carrizosa. A unified approach to extract interpretable rules from tree ensembles via integer programming. *Comput. Oper. Res.*, 185:107283, 2026.

[18] Daniel Borup, Bent Jesper Christensen, Nicolaj Søndergaard Mühlbach, and Mikkel Slot Nielsen. Targeting predictors in random forest regression. *International Journal of Forecasting*, 39(2):841–868, 2023.

[19] Sérgio Branco, André G Ferreira, and Jorge Cabral. Machine learning in resource-scarce embedded systems, fpgas, and end-devices: A survey. *Electronics*, 8(11):1289, 2019.

[20] Leo Breiman. Some infinity theory for predictor ensembles. Technical Report 579, Statistics Department, UC Berkeley, 2000.

[21] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):1–33, 2001.

[22] Leo Breiman. Consistency for a simple model of random forests. Technical Report 670, Statistics Department, UC Berkeley, Berkeley, CA, 2004.

[23] Leo Breiman, Jerome Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Taylor & Francis, Boca Raton, FL, 1984.

[24] Adam D. Bull. Honest adaptive confidence bands and self-similar functions. *Electronic Journal of Statistics*, 6:1490–1516, 2012. doi: 10.1214/12-EJS720.

[25] Sebastian Buschjäger and Katharina Morik. Joint leaf-refinement and ensemble pruning through l1 regularization. *Data Min. Knowl. Discov.*, 37(3):1230–1261, 2023.

[26] Peter Bühlmann and Bin Yu. Boosting with the l2 loss: Regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

[27] A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[28] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[29] Chien-Ming Chi, Patrick Vossler, Yingying Fan, and Jinchi Lv. Asymptotic properties of high-dimensional random forests. *The Annals of Statistics*, 50(6):3415 – 3438, 2022.

[30] Alicia Curth, Alan Jeffares, and Mihaela van der Schaar. A U-turn on double descent: Rethinking parameter counting in statistical learning. In *37th Conference on Neural Information Processing Systems*, 2023.

[31] Alex Davies and Zoubin Ghahramani. The random forest kernel and other kernels for big data from random partitions. *arXiv* preprint, 1402.4293, 2014.

[32] Zh ijie Deng, Jiaxin Shi, and Jun Zhu. Neuralef: Deconstructing kernels by deep neural networks. In *International Conference on Machine Learning*, pages 4976–4992. PMLR, 2022.

[33] Laurens Devos, Timo Martens, Deniz Can Oruc, Wannes Meert, Hendrik Blockeel, and Jesse Davis. Compressing tree ensembles through level-wise optimization and pruning. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 13419–13443, 2025.

[34] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.

[35] David L. Donoho. CART and best-ortho-basis: A connection. *The Annals of Statistics*, 25(5):1870–1911, 1997.

[36] Dheeru Dua and Casey Graff. UCI machine learning repository, 2019. URL `http://archive.ics.uci.edu/ml`.

[37] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[38] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[39] Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916 – 954, 2008.

[40] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv* preprint, 1711.09784, 2017.

[41] Wei Gao and Zhi-Hua Zhou. Towards convergence rate analysis of random forests for classification. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[42] Wei Gao, Fan Xu, and Zhi-Hua Zhou. Towards convergence rate analysis of random forests for classification. *Artificial Intelligence*, 313:103788, 2022.

[43] Robin Genuer. Variance reduction in purely random forests. *J. Nonparametr. Stat.*, 24(3):543–562, 2012.

[44] Evarist Giné and Richard Nickl. Confidence bands in density estimation. *The Annals of Statistics*, 38(2): 1122–1170, 2010. doi: 10.1214/09-AOS738.

[45] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *36th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[46] Peter Grünwald. *The minimum description length principle*. The MIT Press, Cambridge, MA, 2007.

[47] László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag, New York, 2002.

[48] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[49] Trevor Hastie and Robert Tibshirani. *Generalized additive models*. Chapman & Hall, London, 1990.

[50] Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. Generalization bounds: Perspectives from information theory and pac-bayes. *Foundations and Trends in Machine Learning*, 18 (1):1–223, 2025.

[51] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv* preprint, 1503.02531, 2015.

[52] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer Berlin Heidelberg, Berlin, 2004.

[53] Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[54] Alexey Ignatiev, Yacine Izza, Peter J. Stuckey, and Joao Marques-Silva. Using MaxSAT for efficient explanations of tree ensembles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4): 3776–3785, 2022. doi: 10.1609/aaai.v36i4.20292.

[55] Yacine Izza and Joao Marques-Silva. On explaining random forests with sat. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2584–2591, 2021.

[56] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 8571–8580, 2018.

[57] Jason Klusowski. Sparse learning with CART. In *Advances in Neural Information Processing Systems*, volume 33, pages 11612–11622, 2020.

[58] Jason Klusowski. Sharp analysis of a simple model for random forests. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 757–765, 2021.

[59] Jason M. Klusowski and Peter M. Tian. Large scale prediction with decision trees. *Journal of the American Statistical Association*, 119(545):525–537, 2024.

[60] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. Mondrian forests: Efficient online random forests. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[61] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018.

[62] Kaiqiao Li, Sijie Yao, Zhenyu Zhang, Biwei Cao, Christopher M Wilson, Denise Kalos, Pei Fen Kuan, Ruoqing Zhu, and Xuefeng Wang. Efficient gradient boosting for prognostic biomarker discovery. *Bioinformatics*, 38(6):1631–1638, 2022.

[63] Yicheng Li, Zixiong Yu, Guhan Chen, and Qian Lin. On the eigenvalue decay rates of a class of neural-network related kernel functions defined on general domains. *Journal of Machine Learning Research*, 25 (82):1–47, 2024.

[64] Ji Lin, Ligeng Zhu, Wei-Ming Chen, Wei-Chen Wang, and Song Han. Tiny machine learning: Progress and futures [feature]. *IEEE Circuits and Systems Magazine*, 23(3):8–34, 2023.

[65] Brian Liu and Rahul Mazumder. Forestprune: Compact depth-pruned tree ensembles. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 9417–9428, 2023.

[66] Brian Liu and Rahul Mazumder. Fire: An optimization approach for fast interpretable rule extraction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1396–1405, 2023.

[67] Brian Liu, Rahul Mazumder, and Peter Radchenko. Extracting interpretable models from tree ensembles: Computational and statistical perspectives. *Journal of the American Statistical Association*, 2026. doi: 10.1080/01621459.2026.2657609. Published online.

[68] Stephan S. Lorenzen, Christian Igel, and Yevgeny Seldin. On pac-bayesian bounds for random forests. *Machine Learning*, 108(8):1503–1522, 2019.

[69] Rahul Mazumder and Haoyue Wang. On the convergence of CART under sufficient impurity decrease condition. In *Advances in Neural Information Processing Systems*, volume 36, pages 57754–57782, 2023.

[70] Nicolai Meinshausen. Quantile regression forests. *J. Mach. Learn. Res.*, 7:983–999, 2006.

[71] Nicolai Meinshausen. Node harvest. *The Annals of Applied Statistics*, 4(4):2049–2072, 2010.

[72] Lucas Mentch and Giles Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17(26):1–41, 2016.

[73] Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. Universal consistency and minimax rates for online mondrian forests. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[74] Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. Minimax optimal rates for Mondrian trees and forests. *The Annals of Statistics*, 48(4):2253–2276, 2020.

[75] Eliza O'Reilly and Ngoc Mai Tran. Stochastic geometry to generalize the mondrian process. *SIAM Journal on Mathematics of Data Science*, 4(2):531–552, 2022.

[76] Eliza O'Reilly and Ngoc Mai Tran. Minimax rates for high-dimensional random tessellation forests. *Journal of Machine Learning Research*, 25(89):1–32, 2024.

[77] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[78] David Pfau, Stig Petersen, Ashish Agarwal, David GT Barrett, and Kimberly L Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. *arXiv preprint arXiv:1806.02215*, 2018.

[79] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, 13(13):389–427, 2012.

[80] Abhishake Rastogi and Peter Mathé. Inverse learning in hilbert scales. *Machine Learning*, 112(7): 2469–2499, 2023.

[81] Jorma Rissanen. *Stochastic complexity in statistical inquiry*. World Scientific, Singapore, 1989.

[82] J Jon Ryu, Xiangxiang Xu, HS Erol, Yuheng Bu, Lizhong Zheng, and Gregory W Wornell. Operator svd with neural networks via nested low-rank approximation. *arXiv preprint arXiv:2402.03655*, 2024.

[83] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, Cambridge, MA, 2012.

[84] Erwan Scornet. Random forests and kernel methods. *IEEE Trans. Inf. Theory*, 62(3):1485–1500, 2016.

[85] Erwan Scornet and Giles Hooker. Theory of Random Forests: A Review. Technical Report, HAL, 2025.

[86] Erwan Scornet, Gerard Biau, and Jean Philippe Vert. Consistency of random forests. *Ann. Statist.*, 43(4): 1716–1741, 2015.

[87] Clayton Scott and Robert D. Nowak. Minimax-optimal classification with dyadic decision trees. *IEEE Transactions on Information Theory*, 52(4):1335–1353, 2006.

[88] Shahab Shoar, Nicholas Chileshe, and John David Edwards. Machine learning-aided engineering services' cost overruns prediction in high-rise residential building projects: Application of random forest regression. *Journal of Building Engineering*, 50:104102, 2022.

[89] Yan Shuo Tan, Abhineet Agarwal, and Bin Yu. A cautionary tale on fitting decision trees to data from additive models: generalization lower bounds. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 9663–9685, 2022.

[90] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Inf. Fusion*, 81: 84–90, 2022.

[91] Charles J Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10:1040–1053, 1982.

[92] Vasilis Syrgkanis and Manolis Zampetakis. Estimation and inference with trees and forests in high dimensions. In *Proceedings of Thirty Third Conference on Learning Theory*, pages 3453–3454, 2020.

[93] Botond Szabó, A. W. van der Vaart, and J. H. van Zanten. Frequentist coverage of adaptive nonparametric Bayesian credible sets. *The Annals of Statistics*, 43(4):1391–1428, 2015. doi: 10.1214/14-AOS1270.

[94] Cheng Tang, Damien Garreau, and Ulrike von Luxburg. When do random forests fail? In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[95] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

[96] Grigorios Tsoumakas, Ioannis Partalas, and Ioannis Vlahavas. An ensemble pruning primer. In Oleg Okun and Giorgio Valentini, editors, *Applications of Supervised and Unsupervised Ensemble Methods*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[97] Alexandre Tsybakov. *Introduction to Nonparametric Estimation*. Springer, New York, 2008.

[98] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2013.

[99] Thibaut Vidal and Maximilian Schiffer. Born-again tree ensembles. In *International conference on machine learning*, pages 9743–9753. PMLR, 2020.

[100] Binh Duc Vu, Jan Kapar, Marvin Wright, and David S. Watson. Autoencoding random forests. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5487.

[101] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *J. Am. Stat. Assoc.*, 113(523):1228–1242, 2018.

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

[103] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2001.

[104] Marvin N. Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *J. Stat. Softw.*, 77(1), 2017.

[105] Zineng Xu, Subhro Ghosh, and Yan Shuo Tan. On the statistical optimality of optimal decision trees. *arXiv* preprint, 2603.05340, 2026.

[106] Seungmin Yoo and Chanyoung Oh. Microforest: Lightweight bottleneck prediction for manufacturing processes on edge devices. *Applied Sciences*, 15(14):7798, 2025.

[107] Sichong Zhang, Xiong Wang, and Fei Lu. Minimax rates for learning kernels in operators. *arXiv* preprint, 2502.20368, 2025.

[108] Tong Zhang and Bin Yu. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33(4):1538 – 1579, 2005.

[109] Yuqian Zhang, Weijie Ji, and Jelena Bradic. Adaptive split balancing for optimal random forest. *arXiv* preprint, 2402.11228, 2024.

[110] Qin-Cheng Zheng, Shen-Huan Lyu, Shao-Qun Zhang, Yuan Jiang, and Zhi-Hua Zhou. On the consistency rate of decision tree learning algorithms. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 7824–7848, 2023.

[111] Yichen Zhou and Giles Hooker. Interpreting models via single tree approximation. *arXiv preprint arXiv:1610.09036*, 2016.

[112] Yichen Zhou and Giles Hooker. Boulevard: Regularized stochastic gradient boosted trees and their limiting distribution. *Journal of Machine Learning Research*, 23(183):1–44, 2022.

# A Proofs

We restate our assumptions and Thm. 4.1 for completeness. Throughout, let $\boldsymbol{G}_m$ denote the orthogonal projector in $L^2(\mu_X)$ onto the constant function 1 together with the first $m$ nonconstant eigenfunctions $\{\psi_1, \dots, \psi_m\}$ of $\boldsymbol{K}_\Pi$, ordered by decreasing eigenvalue.

(A1) *Bounds*. The feature space is compact on the unit hypercube $\mathcal{X} = [0,1]^d$, with marginal density $\mu_X$ bounded away from 0 and $\infty$, and outcomes on the unit interval $\mathcal{Y} = [0,1]$.

(A2) *Reference operator*. For fixed $\boldsymbol{K}_\Pi := \mathbb{E}_{T\sim\Pi}\big[\boldsymbol{P}_T\big]$, there exist $\alpha \in (0,1]$, $\zeta \in (0,\infty)$, and $g \in L^2(\mu_X)$ with $\|g\|_{L^2} \le \zeta$ such that $f^* = \boldsymbol{K}_\Pi^\alpha g$ and $f^*(\boldsymbol{x}) \in [0,1]$ $\mu_X$-a.e.

(A3) *Polynomial decay*. The operator $\boldsymbol{K}_\Pi$ has nonconstant eigenvalues $\lambda_i \asymp i^{-\beta}$, for some $\beta > 1$. We write $s := 2\alpha\beta$ for the effective spectral smoothness exponent.

(A4) *Spectral localization*. For all $m \ge 1$,

$$\sup_{\omega\in\{-1,1\}^m} \Big\| \sum_{j=m+1}^{2m} \omega_{j-m}\psi_j \Big\|_\infty \lesssim m^{\alpha\beta+1/2}.$$

(A5) *Subsampling*. Trees are trained on subsamples of size $N^{(b)} = aN$ for some fixed $a \in (0,1]$.

(A6) *Budget*. Splits are selected on a best-first basis until a predetermined leaf budget is spent.

(A7) *Tiebreak*. If every candidate split has zero gain, the algorithm splits the largest-mass leaf at the midpoint of a coordinate drawn uniformly at random.

(A8) *Amortized spectral gain*. Fix $\varepsilon \in (0,1)$. There exist constants $c \in (0,1)$ and $C > 0$ such that, on any linear reachability window $|T| \lesssim m$, for every $m \ge 1$ and $f \in \mathcal{F}$, some $h \lesssim \mathfrak{C}_m(\varepsilon)$ further CART splits satisfy

$$V(T;f) - \mathbb{E}_\nu\big[V(T_{+h};f) \mid T\big] \ge c\,U_m(T;f) - C\,\|(\boldsymbol{I}-\boldsymbol{G}_m)f\|_{L^2}^2. \tag{ASG}$$

**Theorem 4.1** (Minimax optimality). Assume (A1)-(A8), with $s := 2\alpha\beta$ and $\mathfrak{C}_m(\varepsilon) = \mathcal{O}(m)$ for some $\varepsilon \in (0,1)$. Then a RF $\hat{f}_{\mathrm{RF}}$ with oracle CART splits, $B = \infty$ trees, and leaf budget $M \asymp N^{1/(s+1)}$ satisfies

$$\sup_{f^*\in\mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \;\lesssim\; N^{-\frac{s}{s+1}}.$$

This rate is minimax optimal over the source class $\mathcal{F}$. Replacing oracle splits with empirical splits preserves the rate up to a logarithmic factor $\log(Nd)^{\frac{s}{s+1}}$ under an empirical variant of ASG with $M \asymp \big(N/\log(Nd)\big)^{1/(s+1)}$ leaves.

Thm. 4.1 makes three separate claims: (1) RFs grown with oracle CART splits attain the stated upper bound $N^{-s/(s+1)}$; (2) the minimax risk over $\mathcal{F}$ is of exact order $N^{-s/(s+1)}$; and (3) replacing oracle splits with empirical splits preserves the rate up to a logarithmic factor under (eASG) and an adjusted leaf budget. We prove each claim in turn.

## A.1 Population RF upper bound

The population argument has two ingredients: the source condition gives a spectral approximation floor of order $m^{-s}$, while ASG shows that oracle CART reaches that floor using order $m$ leaves in the linear-complexity regime.

**Lemma A.1** (Spectral tail)**.** *For every $f^* = \boldsymbol{K}_\Pi^\alpha g \in \mathcal{F}$ and every $m \geq 1$,*

$$\|(\boldsymbol{I} - \boldsymbol{G}_m) f^*\|_{L^2}^2 \leq \zeta^2 \lambda_{m+1}^{2\alpha} \lesssim m^{-s}.$$

*Proof.* Expand the nonconstant component of $g$ in the eigenbasis of $\boldsymbol{K}_\Pi$. Then

$$(\boldsymbol{I} - \boldsymbol{G}_m) f^* = \sum_{j>m} \lambda_j^\alpha \langle g, \psi_j \rangle \psi_j,$$

so

$$\|(\boldsymbol{I} - \boldsymbol{G}_m) f^*\|_{L^2}^2 \leq \lambda_{m+1}^{2\alpha} \|g\|_{L^2}^2 \leq \zeta^2 \lambda_{m+1}^{2\alpha}.$$

Apply (A3) with $s = 2\alpha\beta$. □

**Lemma A.2** (Block recurrence with slack)**.** *Fix $m \geq 1$ and let $T$ be any reachable state. Suppose that, for some $\sigma_m \geq 0$, a CART block taking $T$ to $T' = T_{+h}$ in $h \lesssim \mathfrak{C}_m(\varepsilon)$ further splits satisfies*

$$V(T; f^*) - \mathbb{E}_\nu[V(T'; f^*) \mid T] \geq c\, U_m(T; f^*) - C\|(\boldsymbol{I} - \boldsymbol{G}_m) f^*\|_{L^2}^2 - \sigma_m. \tag{2}$$

*Then*

$$\mathbb{E}_\nu[V(T'; f^*) \mid T] \leq q\, V(T; f^*) + C_1 m^{-s} + \sigma_m, \qquad q := 1 - \frac{c}{2} < 1, \tag{3}$$

*where $C_1 < \infty$ is uniform in $m, f^*, T$. In particular, under population ASG* (ASG) *this holds with $\sigma_m = 0$.*

**Proposition A.3** (Multiscale CART approximation)**.** *Assume $\mathfrak{C}_m(\varepsilon) = \mathcal{O}(m)$. Suppose that the block recurrence* (3) *holds at every resolution $m \geq 1$ on any linear reachability window $|T| \lesssim m$, with a nondecreasing sequence $\sigma_m \geq 0$ and $h \lesssim \mathfrak{C}_m(\varepsilon)$ further splits per block. Then for every $m \geq 1$ there is a deterministic leaf budget $M_m \lesssim m$ such that*

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}_\nu\Big[\|(\boldsymbol{I} - \boldsymbol{P}_{T_{M_m}}) f^*\|_{L^2}^2\Big] \lesssim m^{-s} + \sigma_m.$$

*In particular, population ASG gives $\sigma_m = 0$ and hence the approximation bound $\mathcal{O}(m^{-s})$.*

*Proof.* Let $m_\ell = 2^\ell$, $\ell = 0, \ldots, L$, with $m_L \asymp m$, and write $q := 1 - c/2 < 1$. We show inductively that, at the end of stage $\ell$,

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}_\nu[V(T; f^*)] \leq A\big(m_\ell^{-s} + \sigma_{m_\ell}\big) \tag{4}$$

for a constant $A$ independent of $\ell$.

Since $0 \leq f^* \leq 1$, we have $V(T; f^*) \leq \|f^*\|_{L^2}^2 \leq 1$, so for $m_0 = 1$ the claim holds after enlarging $A$ if necessary. Suppose (4) holds at stage $\ell - 1$. Since $m_{\ell-1}^{-s} = 2^s m_\ell^{-s}$ and $\sigma_m$ is nondecreasing,

$$\mathbb{E}_\nu[V(T; f^*)] \leq A\big(2^s m_\ell^{-s} + \sigma_{m_\ell}\big)$$

at the beginning of stage $\ell$. Fix the integer $k = k(s, c)$ so that $q^k 2^s < 1$. At stage $\ell$, we apply Lemma A.2 $k$ consecutive times at resolution $m_\ell$. By induction on the stages and the $k$ blocks within stage $\ell$, each invocation starts from a tree of size $|T| \lesssim \sum_{j \leq \ell} m_j \lesssim m_\ell$ and uses $h \lesssim \mathfrak{C}_{m_\ell}(\varepsilon) \lesssim m_\ell$ splits, so every block remains within a fixed linear reachability window $|T| \lesssim m_\ell$ and the total leaf budget satisfies

$$M_m \leq 1 + \sum_{\ell \leq L} \mathcal{O}(m_\ell) \lesssim m.$$

After the $k$ blocks at stage $\ell$, Lemma A.2 gives

$$\mathbb{E}_\nu[V(T; f^*)] \leq q^k A\big(2^s m_\ell^{-s} + \sigma_{m_\ell}\big) + \big(C_1 m_\ell^{-s} + \sigma_{m_\ell}\big) \sum_{j=0}^{k-1} q^j.$$

Since $q^k 2^s < 1$ and $q^k < 1$, choosing $A$ sufficiently large absorbs both terms on the right-hand side and closes the induction.

Finally, if the target approximation is reached before the deterministic budget is exhausted, continue splitting arbitrarily. Recursive refinement cannot increase $V(T; f^*)$, so the same bound holds at the predetermined budget $M_m$. □

**Lemma A.4** (Leaf estimation on an oracle partition)**.** *Let $T$ be a recursive partition independent of the sample used to estimate its leaf values, with $|T| \le M$. Let $\hat{f}_T$ be the usual leaf-mean regressogram based on $n$ observations with bounded response $Y \in [0,1]$. Then*

$$\mathbb{E}\Big[\|\hat{f}_T - f^*\|_{L^2}^2\Big] \lesssim \|(\boldsymbol{I} - \boldsymbol{P}_T) f^*\|_{L^2}^2 + \frac{M}{n}.$$

*Proof.* Conditional expectation onto the leaf $\sigma$-algebra gives the orthogonal bias term $\|(\boldsymbol{I} - \boldsymbol{P}_T) f^*\|_{L^2}^2$. For a leaf $A \in T$, write $p_A := \mu_X(A)$ and let $N_A$ denote its sample count. Conditional on $N_A > 0$, boundedness of $Y$ gives

$$\mathbb{E}\big[(\hat{\mu}_A - \mu_A)^2 \mid N_A\big] \le \frac{1}{4N_A}.$$

Since $N_A \sim \mathrm{Binomial}(n, p_A)$,

$$p_A \, \mathbb{E}\left[\frac{\mathbf{1}\{N_A > 0\}}{N_A}\right] \lesssim \frac{1}{n}.$$

If $N_A = 0$, any fixed fallback in $[0,1]$ has squared error at most one, and its integrated contribution is bounded by

$$p_A \mathbb{P}(N_A = 0) = p_A (1 - p_A)^n \le \frac{1}{en}.$$

Thus each leaf contributes $\mathcal{O}(n^{-1})$ to the estimation error. Summing over at most $M$ leaves and adding the orthogonal projection bias proves the result. □

We now prove our first claim.

**Claim 1**. Fix $\varepsilon \in (0,1)$ and assume (A1)–(A8) with $\mathfrak{C}_m(\varepsilon) = \mathcal{O}(m)$. Then a RF $\hat{f}_{\mathrm{RF}}$ with oracle CART splits, $B = \infty$ trees, and leaf budget $M \asymp N^{1/(s+1)}$ satisfies

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \;\lesssim\; N^{-\frac{s}{s+1}}.$$

*Proof.* Take $m \asymp N^{1/(s+1)}$ and the corresponding deterministic budget $M_m \lesssim m$ from Prop. A.3. Because recursive refinement cannot increase the population projection error $V(T; f^*)$, we may pad the tree after the multiscale blocks finish by arbitrary recursive splits to reach any predetermined leaf budget $M \asymp m \asymp N^{1/(s+1)}$. Under (A5), each tree uses $N^{(b)} = aN$ observations, so Lemma A.4 and Prop. A.3 give

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{T_M} - f^*\|_{L^2}^2\Big] \lesssim m^{-s} + \frac{M}{N} \asymp m^{-s} + \frac{m}{N}.$$

For $B = \infty$, Jensen's inequality shows that averaging over tree randomization cannot increase squared $L^2$ risk. Therefore

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \lesssim m^{-s} + \frac{m}{N}.$$

Balancing the two terms at $m \asymp N^{1/(s+1)}$ yields

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \lesssim N^{-s/(s+1)}.$$

□

*Remark* A.5 (Relation to classical tree rates)*.* The dimension-free exponent in Claim 1 is a consequence of the linear spectral tree-complexity regime. More generally, if resolution $m$ incurs approximation error $m^{-s}$ but requires $\mathfrak{C}_m(\varepsilon) \asymp m^\gamma$ leaves, the same bias–variance calculation (formalized in Prop. B.19) gives

$$\mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \lesssim m^{-s} + \frac{m^\gamma}{N},$$

and hence the rate $N^{-s/(s+\gamma)}$. This recovers a familiar non-spectral phenomenon for additive regression. For a model with $d_0 \le d$ active $C^1$ univariate components, resolution $r$ gives squared approximation error of order $r^{-2}$, while a single leaf-average axis-aligned partition may require order $r^{d_0}$ cells to resolve all coordinate directions simultaneously. Balancing $r^{-2} + r^{d_0}/N$ yields $N^{-2/(d_0+2)}$, the sharp tree rate obtained by Shuo Tan et al. [89] for honest leaf-average trees on $C^1$ additive models (corresponding to $s = 2$ and $\gamma = d_0$; see Appx. B.6 and Appx. C for further discussion).

### A.2 Minimax optimality

Because population CART splits depend on the unknown target $f^*$, the oracle forest $\hat{f}_{\mathrm{RF}}$ in Claim 1 is not an ordinary data-only estimator. To establish that $N^{-s/(s+1)}$ is the exact minimax rate over $\mathcal{F}$, we combine a standard Assouad hypercube lower bound [97] using (A2)–(A4) with a simple data-driven spectral projection upper bound.

**Proposition A.6** (Spectral Assouad packing)**.** *Under (A2)–(A4), there are constants $a_0 > 0$ and $\theta \in (0, 1/2]$, depending only on the fixed class parameters, such that for every $m$ the functions*

$$f_\omega := \theta + a_m \sum_{j=m+1}^{2m} \omega_{j-m}\psi_j, \qquad \omega \in \{-1,1\}^m, \qquad a_m := a_0 m^{-(\alpha\beta+1/2)},$$

*belong to $\mathcal{F}$ and take values in a fixed interval $[\delta, 1-\delta] \subset (0,1)$ independent of $m$.*

*Proof.* Because $\boldsymbol{P}_T 1 = 1$ for every $T$, one has $\boldsymbol{K}_\Pi 1 = 1$ and hence $\boldsymbol{K}_\Pi^\alpha 1 = 1$. Choose

$$\theta := \min\left\{\frac{1}{4}, \frac{\zeta}{4}\right\}.$$

The source preimage of $f_\omega$ is

$$g_\omega = \theta + a_m \sum_{j=m+1}^{2m} \lambda_j^{-\alpha}\omega_{j-m}\psi_j.$$

By orthogonality and (A3),

$$\begin{aligned}\|g_\omega\|_{L^2}^2 &= \theta^2 + a_m^2 \sum_{j=m+1}^{2m} \lambda_j^{-2\alpha} \\ &\lesssim \theta^2 + a_0^2 m^{-(2\alpha\beta+1)} \cdot m \cdot m^{2\alpha\beta} \\ &\le \theta^2 + C a_0^2.\end{aligned}$$

Choose $a_0$ sufficiently small that this is at most $\zeta^2$. By (A4),

$$\left\| a_m \sum_{j=m+1}^{2m} \omega_{j-m}\psi_j \right\|_\infty \lesssim a_0.$$

Reducing $a_0$ once more makes this quantity at most $\theta/2$. Hence every $f_\omega$ lies in $[\theta/2, 3\theta/2] \subset (0,1)$, uniformly in $m$. Take $\delta := \theta/2$. □

We may now prove our second claim.

**Claim 2**. Assume (A1)–(A4). Then the minimax rate over the source class $\mathcal{F}$ is given by

$$\inf_{\tilde{f}} \sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\tilde{f} - f^*\|_{L^2}^2\Big] \asymp N^{-s/(s+1)}.$$

*Proof.* For the lower bound, restrict attention to the subexperiment

$$Y \mid \boldsymbol{X} = \boldsymbol{x} \sim \mathrm{Bernoulli}(f_\omega(\boldsymbol{x})), \qquad \omega \in \{-1,1\}^m,$$

with $f_\omega$ as in Prop. A.6. For neighboring vertices $\omega, \omega'$ differing in one coordinate,

$$\|f_\omega - f_{\omega'}\|_{L^2}^2 = 4a_m^2.$$

Since all success probabilities lie in the fixed interval $[\delta, 1-\delta]$, the Bernoulli KL divergence is quadratically bounded, so

$$\mathrm{KL}\big(P_\omega^{\otimes N} \,\|\, P_{\omega'}^{\otimes N}\big) \lesssim N a_m^2.$$

Choose $m \asymp N^{1/(s+1)}$. Because $a_m^2 \asymp m^{-(2\alpha\beta+1)} = m^{-(s+1)}$, neighboring KL divergences are then bounded by an absolute constant. Assouad's lemma therefore gives

$$\inf_{\tilde f} \sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\tilde f - f^*\|_{L^2}^2\Big] \gtrsim m a_m^2 \asymp m^{-s} \asymp N^{-s/(s+1)}.$$

For the matching minimax upper bound, consider the truncated orthogonal-series estimator based on $\mathcal{D}_N = \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^N$. Writing $\psi_0 \equiv 1$ and $\theta_j := \langle f^*, \psi_j \rangle_{L^2(\mu_X)}$ for $j \geq 0$, define

$$\hat\theta_j := \frac{1}{N}\sum_{i=1}^N Y_i \psi_j(\boldsymbol{X}_i), \qquad \hat f_m := \sum_{j=0}^m \hat\theta_j \psi_j.$$

Since $\mathbb{E}[Y_i \psi_j(\boldsymbol{X}_i)] = \theta_j$, $Y_i \in [0,1]$, and $\|\psi_j\|_{L^2(\mu_X)} = 1$, we have

$$\mathbb{E}\big[(\hat\theta_j - \theta_j)^2\big] = \frac{1}{N}\operatorname{Var}\big(Y_1 \psi_j(\boldsymbol{X}_1)\big) \leq \frac{1}{N}\mathbb{E}\big[\psi_j(\boldsymbol{X}_1)^2\big] = \frac{1}{N}.$$

By Parseval's identity and Lemma A.1,

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat f_m - f^*\|_{L^2}^2\Big] = \sup_{f^* \in \mathcal{F}} \left\{ \sum_{j=0}^m \mathbb{E}\big[(\hat\theta_j - \theta_j)^2\big] + \|(\boldsymbol{I} - \boldsymbol{G}_m) f^*\|_{L^2}^2 \right\} \lesssim \frac{m}{N} + m^{-s}.$$

Choosing $m \asymp N^{1/(s+1)}$ yields a genuine data-based estimator achieving risk $\mathcal{O}(N^{-s/(s+1)})$, completing the proof of Claim 2. □

### A.3 Empirical splits

For empirical CART, each tree is trained on a subsample $\mathcal{D}_n$ of size $n := N^{(b)} = aN$ drawn from $\mathcal{D}_N$ under (A5), and thus marginally $\mathcal{D}_n \sim \mu^{\otimes n}$. Because the selected partition depends on the same subsample $\mathcal{D}_n$ used to fit its leaf values, and empirical gain maximization need not follow the population CART trajectory, we require a high-probability analogue of ASG that controls the *population* progress made by the empirically chosen splits.

For $u \geq 1$, write

$$\Lambda_N(u) := 1 + \log(Nd) + u, \qquad \Lambda_N := 1 + \log(Nd).$$

For empirical CART, we assume the following high-probability analogue of (A8) on a generic tree's training subsample $\mathcal{D}_n$: for every $f^* \in \mathcal{F}$ and $u \geq 1$, with probability at least $1 - \exp(-u)$ over $\mathcal{D}_n$, simultaneously for every $m \geq 1$ and reachable $T$ with $|T| \lesssim m$, some $h \lesssim \mathfrak{C}_m(\varepsilon)$ further empirical CART splits satisfy

$$V(T; f^*) - \mathbb{E}_\nu[V(T_{+h}; f^*) \mid T, \mathcal{D}_n] \geq c\, U_m(T; f^*) - C\|(\boldsymbol{I} - \boldsymbol{G}_m) f^*\|_{L^2}^2 - C_{\mathrm{emp}} \frac{\mathfrak{C}_m(\varepsilon)\Lambda_N(u)}{N}, \tag{eASG}$$

where $\nu$ here denotes the tree-level randomization conditional on $\mathcal{D}_n$. The final term in (eASG) is a block-level stochastic penalty of the order expected from uniform concentration over a class of recursive trees with $\mathcal{O}\big(\mathfrak{C}_m(\varepsilon)\big)$ leaves, rather than a per-split error accumulated across the block.

**Lemma A.7** (Uniform adaptive-tree oracle bound)**.** *Assume (A1), with $\mathcal{D}_n \sim \mu^{\otimes n}$ of size $n = aN$. For a recursive partition $T$, let $\hat f_T$ denote the empirical MSE minimizer on $\mathcal{D}_n$ over functions constant on the leaves of $T$, clipped to $[0,1]$ and assigned any fixed value in $[0,1]$ on an empirically empty leaf. There is a universal constant $C_0 < \infty$ such that, for every $M \leq n$ and $u \geq 1$, with probability at least $1 - \exp(-u)$ over $\mathcal{D}_n$, simultaneously for every recursive partition $T$ with $|T| \leq M$,*

$$\|\hat f_T - f^*\|_{L^2}^2 \leq 3\|(\boldsymbol{I} - \boldsymbol{P}_T) f^*\|_{L^2}^2 + C_0 \frac{M \log(Nd) + u}{N}.$$

*Consequently, the same inequality holds when $T$ is selected measurably from $\mathcal{D}_n$.*

*Proof.* Write

$$\widehat{R}(f) := \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2, \qquad \mathcal{E}(f) := \|f - f^*\|_{L^2}^2,$$

and define the empirical excess risk on $\mathcal{D}_n$ by

$$\widehat{\mathcal{E}}(f) := \widehat{R}(f) - \widehat{R}(f^*).$$

Let $\varepsilon_i := y_i - f^*(\boldsymbol{x}_i)$. Since $y_i, f^*(\boldsymbol{x}_i) \in [0,1]$, the variables $\varepsilon_i$ are conditionally mean-zero and uniformly sub-Gaussian.

Set

$$\delta_{M,u} := \frac{M\log(nd) + u}{n} \lesssim \frac{M\log(Nd) + u}{N},$$

where the second relation uses $n = aN$ with fixed $a \in (0,1]$. We use the self-normalized uniform concentration bounds of Xu et al. [105]. Their Lemma 7.2, together with the empirical partition count of Lemma 2.1 and the empirical-equivalence argument of their Lemma E.3, implies that with probability at least $1 - \exp(-u)$ over $\mathcal{D}_n$, simultaneously for every tree function $f$ with at most $M$ leaves and values in $[0,1]$,

$$\left|\widehat{\mathcal{E}}(f) - \mathcal{E}(f)\right| \le C\left\{\|f - f^*\|_{L^2}\sqrt{\delta_{M,u}} + \delta_{M,u}\right\}, \tag{5}$$

for a universal constant $C$. Indeed,

$$\widehat{\mathcal{E}}(f) = \|f - f^*\|_n^2 - 2\langle\varepsilon, f - f^*\rangle_n,$$

and (5) follows by applying their self-normalized quadratic- and multiplier-process bounds to $f - f^*$; note that $\|f - f^*\|_\infty \le 1$.

Applying Young's inequality with the fixed choice $\gamma = 1/2$ gives, after adjusting constants,

$$\left|\widehat{\mathcal{E}}(f) - \mathcal{E}(f)\right| \le \frac{1}{2}\,\mathcal{E}(f) + C_1\delta_{M,u} \tag{6}$$

simultaneously over the same class.

Fix now any recursive partition $T$ with $|T| \le M$ and let

$$f_T^* := \boldsymbol{P}_T f^*.$$

Since $f^* \in [0,1]$, also $f_T^* \in [0,1]$, and $f_T^*$ belongs to the leaf-constant function class associated with $T$. By definition of $\hat{f}_T$,

$$\widehat{\mathcal{E}}(\hat{f}_T) \le \widehat{\mathcal{E}}(f_T^*).$$

Applying (6) to both functions yields

$$\frac{1}{2}\mathcal{E}(\hat{f}_T) - C_1\delta_{M,u} \le \widehat{\mathcal{E}}(\hat{f}_T) \le \widehat{\mathcal{E}}(f_T^*) \le \frac{3}{2}\mathcal{E}(f_T^*) + C_1\delta_{M,u}.$$

Therefore

$$\mathcal{E}(\hat{f}_T) \le 3\mathcal{E}(f_T^*) + 4C_1\delta_{M,u}.$$

Finally,

$$\mathcal{E}(f_T^*) = \|(\boldsymbol{I} - \boldsymbol{P}_T)f^*\|_{L^2}^2,$$

which proves the displayed bound after absorbing $a^{-1}$ and universal constants into $C_0$. The concentration event is uniform over all recursive partitions with at most $M$ leaves. Hence the result remains valid if $T$ is chosen measurably from the same subsample $\mathcal{D}_n$. □

**Claim 3**. Under the assumptions of Thm. 4.1, replacing oracle splits with empirical splits preserves the minimax polynomial rate up to a logarithmic factor under (eASG).

*Proof.* Fix any $f^* \in \mathcal{F}$, $m \ge 1$, and $u \ge 1$, and consider a generic tree trained on its subsample $\mathcal{D}_n$ of size $n = aN$. On the event where (eASG) holds for $\mathcal{D}_n$, applying Lemma A.2 and Proposition A.3 conditionally on $\mathcal{D}_n$ with

$$\sigma_m = C_{\mathrm{emp}}\frac{\mathfrak{C}_m(\varepsilon)\Lambda_N(u)}{N}$$

and $\mathfrak{C}_m(\varepsilon) = \mathcal{O}(m)$ gives a multiscale budget $M_m \lesssim m$. Choosing a predetermined leaf budget $M \asymp m$ with $M \geq M_m$ and padding $T_{M_m}$ to $T_M$ by arbitrary recursive refinements cannot increase the population projection error $\|(\boldsymbol{I} - \boldsymbol{P}_T)f^*\|_{L^2}^2$, so on the (eASG) event we have

$$\mathbb{E}_\nu\Big[\|(\boldsymbol{I} - \boldsymbol{P}_{T_M})f^*\|_{L^2}^2 \,\Big|\, \mathcal{D}_n\Big] \lesssim m^{-s} + \frac{m\Lambda_N(u)}{N}.$$

Intersect this event with the high-probability event of Lemma A.7 applied at leaf budget $M \asymp m$, which holds simultaneously over all $M$-leaf partitions. By a union bound, the combined event $\mathcal{G}_u$ has probability at least $1 - 2\exp(-u)$ over $\mathcal{D}_n$. On $\mathcal{G}_u$, taking conditional expectation $\mathbb{E}_\nu[\,\cdot \mid \mathcal{D}_n]$ over the internal tree randomization gives

$$\mathbb{E}_\nu\Big[\|\hat{f}_{T_M} - f^*\|_{L^2}^2 \,\Big|\, \mathcal{D}_n\Big] \lesssim m^{-s} + \frac{m\Lambda_N(u)}{N}.$$

Since $\hat{f}_{T_M}$ and $f^*$ take values in $[0,1]$, the squared error is at most 1 on $\mathcal{G}_u^c$. Integrating the exponential tail in $u \geq 1$ therefore yields the unconditional single-tree bound

$$\mathbb{E}\Big[\|\hat{f}_{T_M} - f^*\|_{L^2}^2\Big] \lesssim m^{-s} + \frac{m\Lambda_N}{N}.$$

Finally, by Jensen's inequality, averaging across the $B = \infty$ trees in the forest (over both subsample selection and internal tree randomization) cannot increase squared $L^2$ risk:

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \leq \sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{T_M} - f^*\|_{L^2}^2\Big] \lesssim m^{-s} + \frac{m\Lambda_N}{N}.$$

Balancing the two terms gives

$$M \asymp m \asymp \left(\frac{N}{\Lambda_N}\right)^{1/(s+1)} \asymp \left(\frac{N}{\log(Nd)}\right)^{1/(s+1)},$$

and hence

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|_{L^2}^2\Big] \lesssim \left(\frac{\log(Nd)}{N}\right)^{s/(s+1)}.$$

This proves the third claim, thereby completing our proof of Thm. 4.1. □

# B Theorem scope

In this section, we present auxiliary results that delimit the scope of Thm. 4.1. We consider a $d$-dimensional reference forest, an explicit function class, and a greedy tree algorithm for which every hypothesis of our main theorem holds, and for which the minimax risk is exactly $N^{-s/(s+1)}$. Appx. B.1 constructs the operator and verifies (A3)–(A4) together with $\mathfrak{C}_m(\varepsilon) = \mathcal{O}(m)$. Appx. B.2 shows by counterexample that these spectral conditions do *not* imply ASG, which is why an algorithmic hypothesis is needed. Appx. B.3 gives a structural condition on the target under which ASG follows, and Appx. B.4 shows that the resulting class still carries the full minimax difficulty. Appx. B.5 establishes that the popular structural impurity decrease (SID) assumption implies ASG pointwise. Appx. B.6 treats superlinear $\mathfrak{C}_m$ and connects to the approximation-theoretic literature. Throughout this section, $\mu_X$ is presumed uniform on $\mathcal{X} = [0,1]^d$.

## B.1 A canonical $d$-dimensional reference forest

**Definition B.1** (Cyclic dyadic hierarchy)**.** Let $\mathcal{H}_d$ be the infinite rooted binary tree of cells obtained by bisecting every cell at depth $\ell$ at its midpoint along coordinate $j(\ell) = 1 + (\ell \bmod d)$. We write $v$ for a node, $A_v \subseteq \mathcal{X}$ for its cell, $\mathrm{dep}(v)$ for its depth, $w \succ v$ for strict descendants, and $w \succeq v$ for $\{v\} \cup \{w \succ v\}$. There are $2^\ell$ nodes at depth $\ell$ and $|A_v| = 2^{-\mathrm{dep}(v)}$.

For each $v$ let $\psi_v$ be the associated Haar contrast: $\psi_v$ is supported on $A_v$, takes the values $\pm|A_v|^{-1/2}$ on the two children of $A_v$, and satisfies $\|\psi_v\|_{L^2} = 1$, $\int \psi_v \, d\mu_X = 0$. The system $\{1\} \cup \{\psi_v\}_{v \in \mathcal{H}_d}$ is an orthonormal basis of $L^2(\mu_X)$.

**Definition B.2** (Pruned reference forest)**.** Fix $\beta > 1$ and set $p := 2^{-\beta}$. Let $\Pi_\beta$ be the law of the random recursive partition $T$ obtained by splitting the root with probability $p$ and, independently for every node whose parent has been split, splitting that node with probability $p$. Put $\boldsymbol{K}_\Pi := \mathbb{E}_{T\sim\Pi_\beta}[\boldsymbol{P}_T]$.

Although every partition is a pruning of the same underlying dyadic hierarchy, $\Pi_\beta$ is supported on mutually incomparable partitions, since different branches may terminate at different depths. Thus the reference forest is not merely a mixture over truncation depths of a single nested grid sequence.

Because the nonconstant eigenvalues below have multiplicity within each depth, throughout this example we use the canonical Haar eigenbasis $\{\psi_v\}$ ordered by depth (with arbitrary ordering within a depth). Let $\boldsymbol{G}_m$ denote the orthogonal projector onto the constant eigenfunction together with the first $m$ nonconstant Haar modes. Since $\boldsymbol{P}_T 1 = 1$ for every tree, including the constant changes neither CART gains nor approximation error, but ensures that $\|(\boldsymbol{I} - \boldsymbol{G}_m)f\|_{L^2}$ is the genuine nonconstant spectral tail.

**Proposition B.3** (Spectrum, trace, and localization)**.** *Under Defs. B.1–B.2:*

*(i)* $\boldsymbol{K}_\Pi \psi_v = p^{\,\mathrm{dep}(v)+1}\psi_v$ *for every* $v$*, and* $\boldsymbol{K}_\Pi 1 = 1$*;*
*(ii) ordering the nonconstant eigenvalues decreasingly,* $\lambda_j \asymp j^{-\beta}$*, so* (A3) *holds;*
*(iii)* $\operatorname{tr} \boldsymbol{K}_\Pi = \mathbb{E}_{\Pi_\beta}|T| < \infty$ *if and only if* $\beta > 1$*;*
*(iv)* (A4) *holds: for every* $m$*,* $\sup_{\omega\in\{-1,1\}^m} \|\sum_{j=m+1}^{2m} \omega_{j-m}\psi_j\|_\infty \lesssim \sqrt{m} \le m^{\alpha\beta+1/2}$.

*Proof.* (i) Fix $v$ and $T \sim \Pi_\beta$. If $v$ is split in $T$ then both children of $A_v$ are unions of cells of $T$, so $\psi_v$ is $\sigma_T$-measurable and $\boldsymbol{P}_T\psi_v = \psi_v$. If $v$ is not split in $T$ then $A_v$ is contained in a cell $A$ of $T$ on which $\psi_v$ has mean zero, whence $\boldsymbol{P}_T\psi_v = 0$. Splitting $v$ requires splitting $v$ and all its $\mathrm{dep}(v)$ ancestors, which by independence has probability $p^{\mathrm{dep}(v)+1}$. Averaging gives the claim; $\boldsymbol{P}_T 1 = 1$ for every $T$ gives the second.

(ii) By (i) the eigenvalue at depth $\ell$ is $p^{\ell+1} = 2^{-\beta(\ell+1)}$ with multiplicity $2^\ell$. Hence $\#\{j : \lambda_j \ge 2^{-\beta(L+1)}\} = \sum_{\ell\le L} 2^\ell = 2^{L+1} - 1$. If $\lambda_j = 2^{-\beta(\ell+1)}$ then $2^\ell \le j \le 2^{\ell+1}$, so $2^{-\beta}j^{-\beta} \le \lambda_j \le 2^{-\beta}(j/2)^{-\beta}$.

(iii) Since 1 is an eigenfunction with eigenvalue 1,

$$\operatorname{tr} \boldsymbol{K}_\Pi = 1 + \sum_v \lambda_v = 1 + \sum_{\ell\ge 0} 2^\ell p^{\ell+1} = 1 + p\sum_{\ell\ge 0} 2^{(1-\beta)\ell},$$

which is finite iff $\beta > 1$. Likewise $\mathbb{E}_\Pi|T| = 1 + p\sum_{\ell\ge0}(2p)^\ell$, and $\operatorname{tr} \boldsymbol{K}_\Pi = \mathbb{E}_\Pi \operatorname{tr} \boldsymbol{P}_T = \mathbb{E}_\Pi|T|$ because $\boldsymbol{P}_T$ is a projection of rank $|T|$.

(iv) The indices $j \in (m, 2m]$ correspond to nodes at depths $\ell$ and possibly $\ell + 1$, where $2^\ell \asymp m$. Nodes at a common depth have disjoint cells and $\|\psi_v\|_\infty = 2^{\mathrm{dep}(v)/2}$. A signed sum of functions with disjoint supports has sup-norm equal to the largest summand, so the sum over each of the at most two depths is $\lesssim 2^{(\ell+1)/2} \lesssim \sqrt{m}$; add the two. □

*Remark* B.4. Part (iii) shows that $\beta > 1$ in (A3) is exactly the subcriticality condition under which the reference forest has finite expected size. It is a property of the construction, not an extra restriction.

**Proposition B.5** (Linear spectral tree complexity)**.** *For every* $m \ge 1$*,* $\mathfrak{C}_m(0) \le 2m + 1$*. Moreover, for every fixed* $\varepsilon \in (0, 1)$*,*

$$m + 1 \le \mathfrak{C}_m(\varepsilon) \le 2m + 1,$$

*so* $\mathfrak{C}_m(\varepsilon) \asymp m$ *for this example.*

*Proof.* Let $L$ be least with $2^{L+1} - 1 \ge m$. By Prop. B.3(i)–(ii), the leading $m$ eigenfunctions are among $\{\psi_v : \mathrm{dep}(v) \le L\}$. Let $T_L$ be the partition obtained by splitting every node of depth $\le L$; it is ancestor-closed, has $2^{L+1} - 1$ internal nodes and hence $2^{L+1}$ leaves, and every $\psi_v$ with $\mathrm{dep}(v) \le L$ is $\sigma_{T_L}$-measurable. Therefore $\boldsymbol{P}_{T_L}u = u$, i.e. $\|\boldsymbol{Q}_{T_L}u\|^2 = 0$, for every $u \in \mathrm{ran}(\boldsymbol{G}_m)$. Minimality of $L$ gives $2^L - 1 < m$, so $|T_L| = 2^{L+1} < 2(m + 1)$.

For the lower bound, let $T$ have at most $m$ leaves. Since $\mathrm{ran}(\boldsymbol{G}_m)$ has dimension $m + 1$ while $\boldsymbol{P}_T$ has rank at most $m$, there is a unit $u \in \mathrm{ran}(\boldsymbol{G}_m)$ with $\boldsymbol{P}_T u = 0$. Hence $\|\boldsymbol{Q}_T u\|^2 = 1$, so

no such tree can achieve tolerance $\varepsilon < 1$ uniformly over the leading spectral space. Therefore $\mathfrak{C}_m(\varepsilon) \geq m + 1$. □

We also record the elementary identity that drives the following results.

**Lemma B.6** (Gains are squared Haar coefficients)**.** *Let $T$ be a partition all of whose cells are cells of $\mathcal{H}_d$, let $v$ be a leaf of $T$, and let the split of $A_v$ be its canonical bisection. Writing $c_w := \langle f, \psi_w \rangle$,*

$$\Gamma_T(v; f) = c_v^2, \qquad \mu_X(A_v)\,\mathrm{Var}\big(f \mid A_v\big) = \sum_{w \succeq v} c_w^2 =: R_v(f), \qquad V(T; f) = \sum_{v \in \partial T} R_v(f),$$

*where $\partial T$ denotes the leaves of $T$.*

*Proof.* The one-step refinement operator $\boldsymbol{Q}_{T,a} = \boldsymbol{P}_{T \oplus a} - \boldsymbol{P}_T$ is the orthogonal projection onto the span of the new stump, which for the canonical bisection of $A_v$ is exactly $\psi_v$; hence $\Gamma_T(v; f) = \|\boldsymbol{Q}_{T,a} f\|^2 = \langle f, \psi_v \rangle^2$. For the second, the residual of $f$ on $A_v$ is $\sum_{w \succeq v} c_w \psi_w$ by orthonormality and the fact that $\{\psi_w\}_{w \succeq v}$ spans the mean-zero functions on $A_v$ measurable with respect to $\mathcal{H}_d$; take squared norms. The third follows because leaves are disjoint. □

**Definition B.7** (Hierarchy-restricted CART)**.** *Dyadic CART* is the best-first CART variant obtained by setting $m_{\mathrm{try}} = d$ and restricting each current leaf $A_v$ to its unique canonical next split in $\mathcal{H}_d$, namely the midpoint bisection along coordinate $j(\mathrm{dep}(v))$. Thus greediness determines which leaf is refined, while the coordinate and threshold of its refinement are fixed by the hierarchy. If all available canonical gains are zero, the largest-mass leaf is refined, with any remaining tie broken at random. Every cell produced by the algorithm is a cell of $\mathcal{H}_d$.

The restriction is part of the hypotheses of the worked example. An unrestricted split can produce cells misaligned with $\mathcal{H}_d$, after which Lemma B.6 no longer applies. Dyadic trees are the setting of much rigorous tree theory [35, 87, 16]; the theorem of the main text is not restricted in this way, and only the worked example below is.

### B.2 Spectral decay does not imply ASG

**Lemma B.8** (Gating)**.** *Fix any block constant $D < \infty$. For every sufficiently large $m$ and every $\eta_0, c_1 > 0$ there is a regression function $f^*$ in the source class of $\boldsymbol{K}_\Pi$, and a state $T$ reachable by dyadic CART with $|T| = \mathcal{O}(\log m)$, such that for all $q \leq Dm$,*

$$V(T; f^*) - \mathbb{E}\big[V(T_{+q}; f^*) \mid T\big] \;<\; \eta_0\, U_m(T; f^*) - c_1 \|(\boldsymbol{I} - \boldsymbol{G}_m) f^*\|^2.$$

*Hence no choice of a uniform $\mathcal{O}(m)$ block budget makes* (A8) *follow from the source and spectral assumptions alone.*

*Proof.* Let $L$ be large enough that the leading $m$ nonconstant modes are contained among depths at most $L$. Choose a node $w$ from this leading block and an unresolved ancestor $v_0$ of $w$. On a sibling branch, first open a path down to depth $L + 1$, and from its terminal node choose a chain $u_1 \succ u_2 \succ \cdots \succ u_H$ of descendants, with $H := \lceil Dm \rceil$. Thus every $u_r$ lies strictly beyond the leading spectral block. Define a centered finite Haar expansion $h$ with

$$c_{v_0} = 0, \qquad c_w = \gamma, \qquad c_{u_r} = \delta \quad (1 \leq r \leq H),$$

and assign positive coefficients of magnitude at most $\delta$ on the already-open sibling path so that dyadic CART reaches the state $T$ in which $v_0$ and $u_1$ are both available. The path has length $\mathcal{O}(L) = \mathcal{O}(\log m)$; all of its coefficients have already been resolved at $T$ and therefore do not enter $V(T; h)$, while at most $\mathcal{O}(\log m)$ of them can lie outside $\mathrm{ran}(\boldsymbol{G}_m)$. Set all remaining coefficients to zero. Because the expansion is finite, $h$ belongs to the domain of $\boldsymbol{K}_\Pi^{-\alpha}$.

Choose any fixed $\theta \in (0, \min\{1, \zeta\})$ and set $f^* = \theta + \rho h$, with $\rho > 0$ sufficiently small that $f^* \in [0, 1]$ and $\|\boldsymbol{K}_\Pi^{-\alpha} f^*\| \leq \zeta$. Translation and positive rescaling do not change any comparison of CART gains.

Take $T$ to be the reachable state just described. By Lemma B.6, the gain that opens the gate $v_0$ is zero, while the available gain on the distracting branch is $\rho^2 \delta^2 > 0$. Best-first dyadic CART therefore

takes the latter. Inductively it descends the distracting chain for all $H$ steps and never opens $v_0$. Consequently, for every $q \le H$,

$$V(T; f^*) - V(T_{+q}; f^*) \le q\rho^2\delta^2 \le H\rho^2\delta^2,$$

whereas $U_m(T; f^*) \ge \rho^2\gamma^2$ and, for $m$ sufficiently large,

$$\|(\boldsymbol{I} - \boldsymbol{G}_m)f^*\|^2 \le (H + C\log m)\rho^2\delta^2 \le 2H\rho^2\delta^2.$$

Choosing $\delta$ so that $H\delta^2(1 + 2c_1) < \eta_0\gamma^2$ yields the claim. □

The mechanism is that an arbitrarily small *positive* gain is preferred to a zero gain indefinitely. The tie-breaking rule (A7) does not intervene, because the candidate gains do not tie. Note also that the total distracting energy $H\delta^2$ can be made arbitrarily small. The obstruction is not the amount of deep energy but the fact that it screens coarse structure from a myopic criterion. Lemma B.8 is therefore the precise reason (A8) is an assumption separate from (A2)–(A4).

### B.3 Hierarchical visibility implies ASG

**Definition B.9** (Hierarchical visibility)**.** For $\kappa > 0$, let $\mathcal{F}_{\mathrm{HV}}(\kappa)$ be the set of $f \in L^2(\mu_X)$ whose Haar coefficients $c_v = \langle f, \psi_v\rangle$ satisfy

$$c_v^2 \;\ge\; \kappa \sum_{w \succ v} c_w^2 \qquad \text{for every } v \in \mathcal{H}_d. \tag{HV}$$

Condition (HV) says that a region's own coarse contrast is never negligible relative to the structure it contains: large-scale signal cannot be hidden entirely behind a zero-contrast ancestor. It imposes nothing on nodes with no descendant energy, and nothing on the *signs* of any coefficient. It is loosely analogous to self-similarity and polished-tail conditions used in adaptive nonparametric inference [44, 93, 24], although the necessity results there concern honest confidence sets rather than minimax estimation, and we claim only the analogy. It excludes pure interactions such as $f = \mathrm{sgn}(x_1 - \frac{1}{2})\,\mathrm{sgn}(x_2 - \frac{1}{2})$, for which the root contrast vanishes while descendant energy does not.

**Proposition B.10** (HV implies ASG)**.** *Let $f \in \mathcal{F}_{\mathrm{HV}}(\kappa)$ and set $\delta := \kappa/(1+\kappa)$. Then for dyadic CART, at every state $T$ whose cells are cells of $\mathcal{H}_d$,*

$$\Gamma^\star(T; f) \;\ge\; \frac{\delta}{|T|}\, V(T; f). \tag{$*$}$$

*Consequently, for every $C_{\mathrm{reach}} \ge 1$ there is $C_{\mathrm{block}} = C_{\mathrm{block}}(\delta, C_{\mathrm{reach}})$ such that* (A8) *holds on $\mathcal{F}_{\mathrm{HV}}(\kappa)$ for every reachable state with $|T| \le C_{\mathrm{reach}} m$ and some $H \le C_{\mathrm{block}} m \lesssim \mathfrak{C}_m(\varepsilon)$ further splits, where the last comparison uses Prop. B.5.*

*Proof.* By (HV) and Lemma B.6, for any leaf $v$, $R_v = c_v^2 + \sum_{w \succ v} c_w^2 \le c_v^2(1 + \kappa^{-1})$, i.e. $c_v^2 \ge \delta R_v$. Taking the leaf with largest $R_v$ and using $V(T; f) = \sum_{v \in \partial T} R_v \le |T| \max_v R_v$ gives $(*)$.

Best-first CART therefore satisfies $V_{t+1} \le (1 - \delta/t)V_t$ for $t = |T|$. Iterating from $t$ to $t + H$,

$$\frac{V_{t+H}}{V_t} \le \prod_{r=t}^{t+H-1}\Big(1 - \frac{\delta}{r}\Big) \le \exp\Big(-\delta \sum_{r=t}^{t+H-1} \frac{1}{r}\Big) \le \Big(\frac{t}{t+H}\Big)^{\delta}.$$

For a state reachable with $t \le C_{\mathrm{reach}} m$ and $H = \lceil C_{\mathrm{block}} m \rceil$ this is at most

$$\big(C_{\mathrm{reach}}/(C_{\mathrm{reach}} + C_{\mathrm{block}})\big)^{\delta},$$

which is below $1 - \eta_0$ for any fixed $\eta_0 < 1$ once $C_{\mathrm{block}} = C_{\mathrm{block}}(\delta, C_{\mathrm{reach}}, \eta_0)$ is large enough. Finally, $f = \boldsymbol{G}_m f + (\boldsymbol{I} - \boldsymbol{G}_m)f$ and the triangle inequality give

$$V(T; f) \ge \tfrac{1}{2} U_m(T; f) - \|(\boldsymbol{I} - \boldsymbol{G}_m)f\|^2,$$

so

$$V(T; f) - V(T_{+H}; f) \;\ge\; \eta_0 V(T; f) \;\ge\; \tfrac{\eta_0}{2} U_m(T; f) - \eta_0\|(\boldsymbol{I} - \boldsymbol{G}_m)f\|^2,$$

which is (ASG) with $c = \eta_0/2$ and $C = \eta_0$. □

Inequality $(*)$ is a SID condition expressed in tree coordinates. (HV) is a convenient coefficient-level sufficient condition for it. In this worked example, HV therefore yields a one-step SID-type inequality, which is stronger than the blockwise progress required by ASG. The important feature for the minimax construction below is that (HV) constrains coefficient magnitudes but not their signs, so a large hypercube of fine-scale alternatives can remain inside the class.

### B.4 A minimax-hard subclass

**Proposition B.11** (Packing)**.** *Fix $s > 0$ and $1 < a < 2^s$, let $L \geq 1$ and $m := 2^L$. Define centered Haar perturbations $h_\omega$ by coefficient magnitudes*

$$|c_v| = \Big(\frac{E_{\mathrm{dep}(v)}}{2^{\mathrm{dep}(v)}}\Big)^{1/2}, \qquad E_{L-r} := A\, a^r\, 2^{-sL} \quad (0 \leq r \leq L), \qquad c_v = 0 \text{ for } \mathrm{dep}(v) > L,$$

*with signs fixed arbitrarily at depths $< L$ and free at depth $L$:*

$$h_\omega := \sum_{\mathrm{dep}(v)<L} c_v \psi_v + a_L \sum_{\mathrm{dep}(v)=L} \omega_v \psi_v, \qquad \omega \in \{-1,1\}^m, \qquad a_L^2 = E_L/2^L.$$

*Let $f_\omega^* := \frac{1}{2} + h_\omega$. For any fixed source radius $\zeta > 1/2$, a constant $A = A(s,a,d,\zeta) > 0$ can be chosen small enough that:*

- *(i) every $f_\omega^*$ satisfies* (HV) *with $\kappa = a - 1$;*
- *(ii) $\sup_\omega \|\boldsymbol{K}_\Pi^{-\alpha} f_\omega^*\|^2 \leq \zeta^2$, so every $f_\omega^*$ lies in the source class of* (A2)*;*
- *(iii) $\sup_\omega \|h_\omega\|_\infty \leq \frac{1}{4}$, and hence $f_\omega^*(x) \in [\frac{1}{4}, \frac{3}{4}]$;*
- *(iv) $a_L^2 \asymp m^{-(s+1)}$ and $\|f_\omega^* - f_{\omega'}^*\|^2 = 4a_L^2 \,\mathrm{Ham}(\omega,\omega')$.*

*Consequently, if $\mathcal{F}_{\mathrm{HV}}$ denotes the intersection of the source class with $\mathcal{F}_{\mathrm{HV}}(a-1)$, then*

$$\inf_{\hat f} \sup_{f^* \in \mathcal{F}_{\mathrm{HV}}} \mathbb{E}\|\hat f - f^*\|^2 \;\gtrsim\; N^{-s/(s+1)}.$$

*Proof.* (i) Fix $v$ at depth $k = L - r$ with $r \geq 1$. Its subtree contains $2^{j-k}$ nodes at each depth $j \in (k, L]$, so

$$\sum_{w \succ v} c_w^2 = 2^{-k} \sum_{j=k+1}^{L} E_j = 2^{-k} A 2^{-sL} \frac{a^r - 1}{a - 1},$$

while $c_v^2 = 2^{-k} A 2^{-sL} a^r$. Their ratio is $a^r(a-1)/(a^r - 1) \geq a - 1$. Nodes at depth $L$ have no descendant energy, and nodes below depth $L$ have neither coefficient nor descendant energy. The condition is uniform in $\omega$ because only magnitudes enter.

(ii) The constant function is an eigenfunction of $\boldsymbol{K}_\Pi$ with eigenvalue 1 and is orthogonal to every Haar mode. By Prop. B.3(i), $\lambda_v^{-2\alpha} \asymp 2^{s\,\mathrm{dep}(v)}$ with $s = 2\alpha\beta$. Therefore

$$\|\boldsymbol{K}_\Pi^{-\alpha} f_\omega^*\|^2 = \frac{1}{4} + \|\boldsymbol{K}_\Pi^{-\alpha} h_\omega\|^2 \lesssim \frac{1}{4} + A \sum_{r=0}^{L} \Big(\frac{a}{2^s}\Big)^r \leq \frac{1}{4} + \frac{CA}{1 - a2^{-s}}.$$

Since $a < 2^s$ and $\zeta > 1/2$, choosing $A$ sufficiently small makes the right-hand side at most $\zeta^2$.

(iii) At any $x$ exactly one node per depth has $x \in A_v$, and $|c_v| \|\psi_v\|_\infty = E_k^{1/2}$. Hence

$$\|h_\omega\|_\infty \leq \sum_{k \leq L} E_k^{1/2} = A^{1/2} 2^{-sL/2} \sum_{r=0}^{L} a^{r/2},$$

which is bounded uniformly in $L$ because $a2^{-s} < 1$, and can be made at most $1/4$ by reducing $A$.

(iv) $a_L^2 = E_L 2^{-L} = A 2^{-sL} 2^{-L} = A m^{-(s+1)}$, and the separation identity follows from orthonormality.

For the final display, take $Y \mid X = x \sim \mathrm{Bernoulli}(f_\omega^*(x))$. Neighbouring vertices differ in one sign, and since all regression parameters lie in $[1/4, 3/4]$, $\mathrm{KL}(P_\omega^{\otimes N} \| P_{\omega'}^{\otimes N}) \lesssim N a_L^2$. Choosing $m \asymp N^{1/(s+1)}$ makes this $\mathcal{O}(1)$, and the standard Assouad hypercube argument gives a lower bound of order $m a_L^2 \asymp m^{-s} \asymp N^{-s/(s+1)}$. □

**Theorem B.12** (Worked example)**.** *Let $\boldsymbol{K}_\Pi$ be as in Def. B.2 with $\beta > 1$, let $\alpha \in (0, 1]$, $s = 2\alpha\beta$, fix $1 < a < 2^s$ and a source radius $\zeta > 1/2$, and let $\mathcal{F}_{\mathrm{HV}}$ be the intersection of the corresponding source class with $\mathcal{F}_{\mathrm{HV}}(a-1)$. Then population dyadic CART forests grown to $M \asymp N^{1/(s+1)}$ leaves satisfy*

$$\inf_{\hat{f}} \sup_{f^* \in \mathcal{F}_{\mathrm{HV}}} \mathbb{E}\Big[\|\hat{f} - f^*\|^2\Big] \asymp \sup_{f^* \in \mathcal{F}_{\mathrm{HV}}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|^2\Big] \asymp N^{-s/(s+1)}.$$

*Proof.* Uniform design and the response model $Y \mid \boldsymbol{X} = \boldsymbol{x} \sim \mathrm{Bernoulli}(f^*(\boldsymbol{x}))$ verify (A1) across $\mathcal{F}_{\mathrm{HV}}$, while Prop. B.11 constructs the class inside (A2). Prop. B.3 gives (A3)–(A4), Prop. B.5 gives $\mathfrak{C}_m(\varepsilon) = \mathcal{O}(m)$, and Prop. B.10 gives (A8) for hierarchy-restricted population CART. The leaf budget in (A6) is built into Def. B.7; the subsampling condition (A5) may be imposed independently when fitting leaf means and does not alter the population split trajectory. Under HV a state with unresolved energy has a strictly positive canonical gain by $(*)$, so the zero-gain convention in (A7) is immaterial before the signal is exhausted. The upper bound of Thm. 4.1 (Claim 1) therefore gives

$$\sup_{f^* \in \mathcal{F}_{\mathrm{HV}}} \mathbb{E}[\|\hat{f}_{\mathrm{RF}} - f^*\|^2] \lesssim N^{-s/(s+1)}.$$

For the matching minimax upper bound over ordinary data-based estimators $\hat{f}$, note that the fixed complete dyadic partition $T_L$ from Prop. B.5 has $|T_L| \asymp m$ leaves independently of $f^*$ and resolves all Haar modes through depth $L$, so its leaf-mean regressogram achieves risk $\lesssim m^{-s} + m/N \asymp N^{-s/(s+1)}$ at $m \asymp N^{1/(s+1)}$ by Lemma A.4.

Finally, Prop. B.11 supplies the minimax lower bound $\gtrsim N^{-s/(s+1)}$ over ordinary estimators $\hat{f}$. Moreover, on the Assouad hypercube $\{f^*_\omega : \omega \in \{-1, 1\}^m\}$ of Prop. B.11, all $f^*_\omega$ share identical squared Haar coefficients $c_v^2$ at every node $v \in \mathcal{H}_d$. By Lemma B.6, the population dyadic CART gains $\Gamma_T(v; f^*_\omega) = c_v^2$ are invariant to $\omega$, so the oracle partition law of $\hat{f}_{\mathrm{RF}}$ is identical across the hypercube. Consequently, $\hat{f}_{\mathrm{RF}}$ is a fixed, $\omega$-independent estimator on this subexperiment, and the Assouad lower bound of Prop. B.11 applies directly to $\hat{f}_{\mathrm{RF}}$ as well. □

*Remark* B.13 (Why the packing has no logarithmic loss). The compatibility window $1 < a < 2^s$ is doing two jobs at once. The inequality $a > 1$ gives a fixed positive visibility constant $\kappa = a - 1$, while $a < 2^s$ makes the source-norm series in Prop. B.11(ii) summable. At the boundary $a = 2^s$ that series grows like $L \asymp \log m$, whereas any strict interior choice keeps both the HV constant and the source radius fixed. Since ASG only requires a fixed blockwise contraction, there is no need to push $a$ to this boundary, and the minimax packing incurs no extra logarithmic factor.

*Remark* B.14 (Interactions). For $d \geq 2$, the class $\mathcal{F}_{\mathrm{HV}}$ is not additive. A depth-$L$ contrast $\psi_v$ splits coordinate $j(L)$ within a cell determined by the preceding $L$ splits, which cycle through all $d$ coordinates. Free signs at depth $L$ therefore produce a contrast in one coordinate whose sign depends on the position in the others, i.e. a genuine multi-way interaction. The construction is not a sum of univariate functions embedded in $\mathbb{R}^d$.

*Remark* B.15 (An algorithmic alternative to HV). $\mathcal{F}_{\mathrm{HV}}$ restricts the function class while leaving CART unchanged. The opposite trade is also available. In Lemma B.8 the distracting gains satisfy $\delta^2 \ll V(T; f)$, so a rule that explores whenever $\max_a \Gamma(a) < \theta V(T)$ — splitting the leaf of largest impurity at the midpoint of a random offered coordinate — opens the gate after $\mathcal{O}(d)$ explorations in expectation. Such a rule would preserve the full source ball at the cost of a further deviation from CART; we do not pursue it here.

### B.5 Structural impurity decrease and amortized spectral gain

Numerous authors have studied variants of the structural impurity decrease (SID) assumption [102, 92, 29, 57], which imposes a uniform lower bound on the population improvement available from a single tree split.

For a measurable rectangle $A \subseteq [0, 1]^d$ with $\mu_X(A) > 0$, define its probability-weighted conditional variance by

$$v_f(A) := \mu_X(A) \operatorname{Var}(f(\boldsymbol{x}) \mid \boldsymbol{x} \in A).$$

Given an admissible axis-aligned split $(j, \tau)$ of $A$, write

$$A_L(j, \tau) := A \cap \{\boldsymbol{x} : x_j \leq \tau\}, \qquad A_R(j, \tau) := A \cap \{\boldsymbol{x} : x_j > \tau\},$$

and define the corresponding population impurity decrease

$$\Delta_f(A; j, \tau) := v_f(A) - v_f(A_L(j, \tau)) - v_f(A_R(j, \tau)).$$

Equivalently, whenever both children have positive mass,

$$\Delta_f(A; j, \tau) = \frac{\mu_X(A_L)\mu_X(A_R)}{\mu_X(A)} \Big( \mathbb{E}\big[f(\boldsymbol{x}) \mid \boldsymbol{x} \in A_L\big] - \mathbb{E}\big[f(\boldsymbol{x}) \mid \boldsymbol{x} \in A_R\big] \Big)^2.$$

**Definition B.16** (Structural impurity decrease)**.** A function $f \in L^2$ satisfies SID with constant $\lambda \in (0, 1]$ if, for every admissible rectangular cell $A$ with $\mu_X(A) > 0$,

$$\sup_{(j,\tau) \in \mathcal{S}(A)} \Delta_f(A; j, \tau) \geq \lambda \, v_f(A),$$

where $\mathcal{S}(A)$ denotes the admissible axis-aligned splits of $A$.

SID has proven useful in analyses of adaptive CART and random forests, though its requirement of a uniform local impurity decrease can be difficult to verify directly [59]. A notable recent result of Mazumder and Wang [69] gives tractable sufficient conditions for SID. In particular, they show that additive regression functions satisfy SID when their component functions obey a local reverse Poincaré inequality, thereby connecting SID to more familiar nonparametric smoothness classes.

In this subsection, we show that SID supplies a simple pointwise sufficient condition for ASG.

**Proposition B.17** (SID implies ASG pointwise)**.** *Suppose $f \in L^2$ satisfies population SID with constant $\lambda > 0$. Suppose further that, conditional on any current leaf, the tree randomization offers a SID-witnessing coordinate with probability at least $\pi_0 > 0$. Then, on any reachability window*

$$|T| \leq C_{\mathrm{reach}} m,$$

*$f$ satisfies the conclusion of ASG with a block budget $h \leq C_0 m$, where $C_0$ depends only on the desired contraction, $C_{\mathrm{reach}}$, $\lambda$, and $\pi_0$.*

*Proof.* For each leaf $A \in T$, write

$$v_f(A) := \mu_X(A) \operatorname{Var}(f(\boldsymbol{x}) \mid \boldsymbol{x} \in A).$$

Since $\boldsymbol{P}_T f$ is the conditional expectation of $f$ on the leaf $\sigma$-algebra,

$$V(T; f) = \|(\boldsymbol{I} - \boldsymbol{P}_T) f\|_{L^2}^2 = \sum_{A \in T} v_f(A).$$

Let $t := |T|$. By SID, a leaf $A$ admits a population split with gain at least $\lambda v_f(A)$. Since a SID-witnessing coordinate is offered with probability at least $\pi_0$, best-first CART has conditional expected gain at least

$$\bar{\lambda} \max_{A \in T} v_f(A), \qquad \bar{\lambda} := \pi_0 \lambda.$$

Hence

$$\mathbb{E}_\nu[V(T_{t+1}; f) \mid T_t] \leq V(T_t; f) - \bar{\lambda} \max_{A \in T_t} v_f(A) \leq \left(1 - \frac{\bar{\lambda}}{t}\right) V(T_t; f).$$

Iterating for $H = \lceil C_0 m \rceil$ further splits gives

$$\mathbb{E}_\nu[V(T_{t+H}; f) \mid T_t] \leq \prod_{r=0}^{H-1} \left(1 - \frac{\bar{\lambda}}{t + r}\right) V(T_t; f) \leq \left(\frac{t}{t + H}\right)^{\bar{\lambda}} V(T_t; f).$$

If $t \leq C_{\mathrm{reach}} m$, then

$$\mathbb{E}_\nu[V(T_{t+H}; f) \mid T_t] \leq \left(\frac{C_{\mathrm{reach}}}{C_{\mathrm{reach}} + C_0}\right)^{\bar{\lambda}} V(T_t; f).$$

Choosing $C_0$ sufficiently large therefore yields a fixed contraction: for some $\eta \in (0, 1)$,

$$V(T_t; f) - \mathbb{E}_\nu[V(T_{t+H}; f) \mid T_t] \geq \eta V(T_t; f).$$

Finally,

$$V(T; f) \geq \frac{1}{2} U_m(T; f) - \|(\boldsymbol{I} - \boldsymbol{G}_m) f\|_{L^2}^2,$$

so

$$V(T; f) - \mathbb{E}_\nu[V(T_{t+H}; f) \mid T] \geq c \, U_m(T; f) - C \|(\boldsymbol{I} - \boldsymbol{G}_m) f\|_{L^2}^2$$

with $c = \eta/2$ and $C = \eta$. Since $H = \mathcal{O}(m)$, this is the ASG conclusion in the linear-complexity regime. □

*Remark* B.18 (Scope of the SID implication). The preceding implication is pointwise in the target $f$: a fixed SID constant gives a simple sufficient condition for amortized progress. ASG is strictly blockwise in character and can accommodate trajectories for which no uniform one-step impurity-decrease constant is available. The quantifiers should therefore be kept distinct when importing SID examples into the spectral source-class setting. In particular, SID for each member of a class does not establish uniform ASG unless the SID constant is itself uniform over that class; spectral superpositions may otherwise create cancellations or gating effects.

### B.6 Superlinear complexity and the approximation-theoretic view

**Proposition B.19** (Rate under polynomial complexity)**.** *Suppose $\mathfrak{C}_m(\varepsilon) \lesssim m^\gamma$ for some $\gamma \geq 1$ and, for this subsection, strengthen ASG so that it holds uniformly over reachable states with $|T| \lesssim m^\gamma$ and uses $h \lesssim \mathfrak{C}_m(\varepsilon)$ further splits. Then the population-split forest satisfies*

$$\sup_{f^* \in \mathcal{F}} \mathbb{E}\Big[\|\hat{f}_{\mathrm{RF}} - f^*\|^2\Big] \lesssim \inf_m \Big\{m^{-s} + \frac{m^\gamma}{N}\Big\} \asymp N^{-s/(s+\gamma)},$$

*attained at $m \asymp N^{1/(s+\gamma)}$. Under the corresponding generalized empirical ASG condition, the empirical-split rate is $\big(\log(Nd)/N\big)^{s/(s+\gamma)}$.*

*Proof.* At dyadic spectral scales up to $m$, the cumulative multiscale budget is

$$\sum_{\ell:\, 2^\ell \leq m} \mathfrak{C}_{2^\ell}(\varepsilon) \lesssim \sum_{\ell:\, 2^\ell \leq m} 2^{\gamma \ell} \lesssim m^\gamma,$$

which is why the enlarged reachability window is required. The same block contraction argument as in Thm. 4.1 therefore gives approximation error $\mathcal{O}(m^{-s})$ using $M \lesssim m^\gamma$ leaves. For population-selected partitions, estimating the leaf means contributes $\mathcal{O}(M/N)$, giving $m^{-s} + m^\gamma/N$; balancing the terms yields the stated choice of $m$ and rate. Under generalized eASG, the adaptive-tree complexity term is $\mathcal{O}(M \log(Nd)/N)$ and the same optimization gives the empirical rate. □

This is an upper bound. A matching *estimator-class* lower bound requires a separate argument: in the sparse additive case with $d_0$ active $C^1$ components ($s = 2, \gamma = d_0$), the lower bound of Shuo Tan et al. [89, Thm. 1] shows that axis-aligned leaf-averaging tree estimators cannot improve on $N^{-2/(d_0+2)}$ (see Remark A.5 and Appx. C). The additive class is therefore a diagnostic rather than an information-theoretic minimax example: additive estimators can achieve $N^{-2/3}$, whereas $\gamma = d_0$ quantifies the representational overhead induced by single-tree geometry.

**Proposition B.20** (Transfer from tree approximation)**.** *If $\sup_{\|g\| \leq \zeta} \inf_{|T| \leq L} \|\boldsymbol{Q}_T \boldsymbol{K}_\Pi^\alpha g\|^2 \lesssim L^{-s}$, then $\mathfrak{C}_m(\varepsilon) \lesssim \varepsilon^{-1/s} m$.*

*Proof.* For a unit $\boldsymbol{u} \in \mathrm{ran}(\boldsymbol{G}_m)$ one has $\|\boldsymbol{K}_\Pi^{-\alpha} \boldsymbol{u}\| \leq \lambda_m^{-\alpha}$, so $\zeta \lambda_m^\alpha \boldsymbol{u}$ lies in the source ball. The hypothesis gives $\inf_{|T| \leq L} \|\boldsymbol{Q}_T \boldsymbol{u}\|^2 \lesssim \lambda_m^{-2\alpha} L^{-s} \asymp (m/L)^s$, and $L = \varepsilon^{-1/s} m$ makes this $\lesssim \varepsilon$. □

Prop. B.20 shows that matching recursive-tree and spectral approximation rates is a simple sufficient condition for linear spectral tree complexity. It provides the bridge to the approximation theory for anisotropic and spatially inhomogeneous classes [1, 105]: results of that type supply exactly the hypothesis above. We do not claim that unrestricted classes of that kind satisfy (A8) — Lemma B.8 shows representation complexity alone cannot imply it — and regard $\mathcal{F}_{\mathrm{HV}}$ as the greedy-accessible analogue.

## C Related works

In this section, we situate Thm. 4.1 within the contemporary literature on convergence rates for decision trees and random forests (see Table 3 for a summary).

**Non-adaptive vs. greedy ensembles.** As noted in the main text, several works establish minimax optimality for *non-adaptive* tree ensembles, including Mondrian forests [60, 73, 74] and random tessellation forests [75, 76]. Because their partitions are drawn independently of the responses $Y$, these analyses reduce to kernel or local-averaging theory on fixed random meshes and do not explain how or why data-driven, greedy splitting succeeds in practice. Closer to the adaptive setting, Zhang et al. [109] obtain minimax rates for modified RFs by enforcing adaptive split balancing (constraining candidate splits to remain geometrically balanced while adapting to smoothness) followed by leaf-wise regressions. By contrast, our focus is on unmodified, best-first CART splitting with ordinary leaf-constant predictions, isolating the spectral and algorithmic conditions under which standard greedy impurity reduction attains minimax risk.

**Globally optimal trees vs. greedy CART.** A second line of work studies *globally optimal* decision trees—estimators that minimize empirical risk over the space of all binary trees with at most $M$ leaves [89, 105]. Notably, Xu et al. [105] establish minimax-optimal rates (up to logarithmic factors) for globally optimal decision trees over broad anisotropic and spatially inhomogeneous function classes. However, computing a globally optimal decision tree is NP-complete in general [53]. We distinguish between such global tree optimizers and greedy CART—both in its *population* form (which accesses exact conditional expectations for candidate splits) and its *empirical* form (which estimates split gains from $\mathcal{D}_N$). Thm. 4.1 shows that greedy CART matches the exact minimax rate in the population regime and the minimax rate up to logarithmic factors in the empirical regime, trading intractable global search over partitions for the amortized spectral gain condition (ASG)/(eASG).

**Adaptive CART and random forests.** Several authors have analyzed greedy CART and random forest variants under structural assumptions on the regression function. Notable examples include:

- Scornet et al. [86] prove the first $L^2$-consistency guarantee for Breiman's original RF algorithm under an additive regression model, showing adaptation to sparse active coordinates without deriving explicit finite-sample rates.
- Syrgkanis and Zampetakis [92] analyze greedy CART with binary features under a sparsity constraint and a structural impurity decrease (SID) condition, obtaining MSE bounds that scale logarithmically in the ambient dimension $d$ and establishing valid asymptotic inference for honest forests.
- Klusowski [57] studies a single tree with cost-complexity pruning under SID and shows that CART adapts to latent low-dimensional structure in sparse additive models at logarithmic depth.
- Chi et al. [29] extend the SID-based analysis to forests built from the empirical CART criterion, deriving high-dimensional convergence rates while allowing for discontinuous $f^*$ and dependent covariates.
- Klusowski and Tian [59] establish universal consistency and explicit high-dimensional rates for CART and RFs on additive models under an $\ell_1$-variation sparsity constraint—even when $d$ grows exponentially in $N$—without imposing a local-gain condition or covariate independence.

Our framework differs from these works in three respects. First, rather than restricting $f^*$ to an additive or binary-feature model, we work on a continuous domain $\mathcal{X} = [0,1]^d$ and define the target class $\mathcal{F}$ directly through the spectral source condition (A2) and eigenvalue decay (A3) of the reference forest operator $\boldsymbol{K}_\Pi$, which naturally accommodates multi-way interactions (Appx. B.4). This inductive bias is arguably more appropriate to tree- and forest-based estimators. Second, whereas Syrgkanis and Zampetakis [92], Klusowski [57], and Chi et al. [29] rely on a one-step SID inequality, our assumption (A8) requires only *blockwise* amortized progress across $\mathcal{O}(\mathfrak{C}_m(\varepsilon))$ splits; as shown in Prop. B.17, pointwise SID is a special case that implies ASG. Third, we establish matching upper and minimax lower bounds ($N^{-s/(s+1)}$) over $\mathcal{F}$.

**When do random forests fail? Gating and Tang et al. [94].** Tang et al. [94] identify structural regimes in which random forests fail to achieve optimal performance, notably when signal is concentrated in high-order interactions whose marginal projections vanish, or when uninformative splits dilute leaf sample sizes. Our gating counterexample in Lemma B.8 provides a sharp spectral formulation of the first phenomenon: even when $f^*$ lies inside the spectral source ball $\mathcal{F}$, a zero coarse contrast ($c_{v_0} = 0$) can act as a "gate" that hides large-scale descendant signal behind an arbitrarily small positive distraction on another branch. This explains why spectral smoothness (A2)–(A4) alone

Table 3: Comparison of theoretical guarantees for decision tree and random forest variants.

| Work | CART basis | Local gain | Target class | Guarantee |
|---|---|---|---|---|
| Mourtada et al. [74] | ✗ | ✗ | Hölder | minimax |
| O'Reilly and Tran [76] | ✗ | ✗ | Hölder | minimax |
| Xu et al. [105] | ✗ | ✗ | Besov | minimax (up to log) |
| Zhang et al. [109] | ✗ | ✗ | Hölder | minimax |
| Scornet et al. [86] | ✓ | ✗ | sparse, additive | consistency |
| Syrgkanis and Zampetakis [92] | ✓ | ✓ | sparse, binary $X$ | convergence rate |
| Chi et al. [29] | ✓ | ✓ | sparse, additive | convergence rate |
| Klusowski [57] | ✓ | ✓ | sparse, additive (1 tree) | rate / adaptivity |
| Klusowski and Tian [59] | ✓ | ✗ | sparse, additive ($\ell_1$ BV) | universal consistency |
| Shuo Tan et al. [89] | ✓ | ✗ | sparse, additive ($C^1$) | tree lower bound |
| **This work (Thm. 4.1)** | ✓ | ✓(ASG) | spectral source class $\mathcal{F}$ | minimax (oracle; log emp.) |

cannot guarantee greedy convergence and why an algorithmic visibility condition such as ASG (A8) or Hierarchical Visibility (Def. B.9) is indispensable.

**Reconciling with the lower bound of Shuo Tan et al. [89].** At first glance, Thm. 4.1 might appear to be in tension with Shuo Tan et al. [89], who prove a "cautionary tale" showing that axis-aligned, leaf-averaging trees *cannot* attain the univariate minimax rate for sparse additive models. This apparent discrepancy dissolves once we distinguish between the ambient additive class and the tree-adapted spectral class $\mathcal{F}$.

Specifically, Shuo Tan et al. [89] consider a sparse additive target $f^*(\boldsymbol{x}) = \sum_{j\in S} \phi_j(x_j)$ with $|S| = d_0$ active coordinates and $C^1$ components. The information-theoretic minimax rate over this additive class is $N^{-2/3}$ (with smoothness $s = 2$ in squared-error scaling), reflecting an effective dimension of one [79]. However, Shuo Tan et al. [89] prove that no single axis-aligned leaf-averaging tree—even with an oracle partition—can improve upon $N^{-2/(d_0+2)}$: piecewise-constant leaf averages cannot exploit additive separability and must instead tensorize across all $d_0$ active coordinates, requiring $M \asymp r^{d_0}$ leaves to achieve a mesh of order $r^{-1}$ in each active direction.

In our framework, this geometric bottleneck is captured directly by the superlinear spectral tree-complexity regime of Remark A.5 and Prop. B.19. Achieving squared approximation error $m^{-s}$ ($s = 2$) simultaneously across $d_0$ active coordinates requires $\mathfrak{C}_m(\varepsilon) \asymp m^\gamma$ leaves with $\gamma = d_0$, which yields the exact oracle tree rate of Shuo Tan et al. [89]:

$$N^{-\frac{s}{s+\gamma}} = N^{-\frac{2}{d_0+2}}.$$

Viewed in terms of the leaf budget $M \asymp m^{d_0}$, a partition that resolves $d_0$ active coordinates to accuracy $M^{-2/d_0}$ also accommodates $d_0$-way hierarchical interactions at that scale. Thus the sparse additive class is strictly contained in a much larger space of $d_0$-dimensional tree-approximable functions, and enlarging the class slows the achievable rate from $N^{-2/3}$ to $N^{-2/(d_0+2)}$. Crucially, this bottleneck arises entirely from the single-tree projection bias $\|(\boldsymbol{I} - \boldsymbol{P}_T)f^*\|_{L^2}^2$—the one term in our risk decomposition that does not shrink with ensemble size $B$ when all trees share the same partition geometry. Consequently, increasing $B$ reduces estimation variance but cannot overcome the representational barrier identified by Shuo Tan et al. [89].

# D Experimental Setup

RFs and GBMs were trained using the `scikit-learn` package. Exact matrix decompositions were computed with `numpy`, while approximate decompositions utilized `scipy`. The SCATE model was implemented within the `torch` framework. Finally, standard Python libraries were used to evaluate metrics such as model size and inference time. All experiments were conducted in one of two setups: locally, on a Macbook Pro with an M5 Pro chip and 24GB memory, or on high performance computing, with 12 AMD EPYC 7282 CPUs, 64GB RAM, and an NVIDIA A30 chip.

While the complexity of our proposed method is independent of the base model's size, the majority of the pruning techniques we benchmark against extract information directly from the ensemble's internal rules and structure. Consequently, employing an overly large model would unfairly increase

the compression difficulty for baseline methods, while offering us an advantage at no cost. To ensure a fair comparison, we standardized the hyperparameters across all RFs and GBMs, aligning them closely with the configurations used in [33]. Specifically, RFs were configured with `n_estimators=250`, `max_depth=15`, while GBMs were trained with `n_estimators=100`, `max_depth=6`. For GBMs, we additionally set `base_score=0` to be correct to the definition of the smoother matrix $\boldsymbol{S}$. We detail the construction of each respective matrix below, in Algorithm 1 and Algorithm 2, for completeness.

**Algorithm 1** Construction of the RF kernel matrix

**Require:** Fitted forest with $B_{\mathrm{RF}}$ trees; reference points $X = (x_1, \ldots, x_n)$; evaluation points $X_* = (x_1^*, \ldots, x_m^*)$.

**Ensure:** Reference kernel $\mathbf{K}_{\mathrm{RF}} \in \mathbb{R}^{n \times n}$ and evaluation kernel $\mathbf{K}_* \in \mathbb{R}^{m \times n}$.

1: Initialize $\mathbf{K}_{\mathrm{RF}} \leftarrow \mathbf{0}_{n \times n}$ and $\mathbf{K}_* \leftarrow \mathbf{0}_{m \times n}$.

2: **for** $b = 1, \ldots, B_{\mathrm{RF}}$ **do**

3: Let $\ell_b(x)$ denote the leaf containing $x$ in tree $b$.

4: For each leaf $\ell$, compute its reference-sample count

$$N_{b,\ell} = \sum_{j=1}^{n} \mathbf{1}\{\ell_b(x_j) = \ell\}.$$

5: Define the normalized leaf weight

$$k_b(x, x_j) = \begin{cases} \dfrac{\mathbf{1}\{\ell_b(x) = \ell_b(x_j)\}}{N_{b,\ell_b(x)}}, & N_{b,\ell_b(x)} > 0, \\ 0, & N_{b,\ell_b(x)} = 0. \end{cases}$$

6: Construct $[\mathbf{K}_b]_{ij} = k_b(x_i, x_j)$ and $[\mathbf{K}_{b,*}]_{ij} = k_b(x_i^*, x_j)$.

7: Set $w_b = 1/B_{\mathrm{RF}}$ for an ordinary RF; for a weighted forest, use its stored tree weight $w_b$.

8: Update

$$\mathbf{K}_{\mathrm{RF}} \leftarrow \mathbf{K}_{\mathrm{RF}} + w_b \mathbf{K}_b,$$
$$\mathbf{K}_* \leftarrow \mathbf{K}_* + w_b \mathbf{K}_{b,*}.$$

9: **end for**

10: **return** $\mathbf{K}_{\mathrm{RF}}, \mathbf{K}_*$.

**Algorithm 2** Construction of the GBM smoother matrix

**Require:** Fitted squared-error GBM with $B_{\mathrm{GBM}}$ ordered trees and learning rate $\eta$; reference points $X = (x_1, \ldots, x_n)$; evaluation points $X_* = (x_1^*, \ldots, x_m^*)$.

**Ensure:** Reference smoother $\mathbf{S}_{\mathrm{GBM}} \in \mathbb{R}^{n \times n}$ and evaluation smoother $\mathbf{S}_* \in \mathbb{R}^{m \times n}$.

1: Initialize $\mathbf{S}_0 \leftarrow \mathbf{0}_{n \times n}$ and $\mathbf{S}_{*,0} \leftarrow \mathbf{0}_{m \times n}$.

2: This initialization corresponds to $f_0(x) = 0$.

3: **for** $b = 1, \ldots, B_{\mathrm{GBM}}$ **do**

4: Construct the normalized leaf-membership matrices $\mathbf{K}_b \in \mathbb{R}^{n \times n}$ and $\mathbf{K}_{b,*} \in \mathbb{R}^{m \times n}$ for tree $b$, as in Algorithm 1.

5: Compute the residual operator

$$\mathbf{R}_{b-1} \leftarrow \mathbf{I}_n - \mathbf{S}_{b-1}.$$

6: Update both smoothers using the same residual operator:

$$\mathbf{S}_b \leftarrow \mathbf{S}_{b-1} + \eta \mathbf{K}_b \mathbf{R}_{b-1},$$
$$\mathbf{S}_{*,b} \leftarrow \mathbf{S}_{*,b-1} + \eta \mathbf{K}_{b,*} \mathbf{R}_{b-1}.$$

7: **end for**

8: Set $\mathbf{S}_{\mathrm{GBM}} \leftarrow \mathbf{S}_{B_{\mathrm{GBM}}}$ and $\mathbf{S}_* \leftarrow \mathbf{S}_{*,B_{\mathrm{GBM}}}$.

9: **return** $\mathbf{S}_{\mathrm{GBM}}, \mathbf{S}_*$.

For our experiments, we assembled small-to-medium datasets from three sources: the UCI Machine Learning Repository [36], OpenML [98], and Kaggle. To equitably demonstrate the performance of all methods, these datasets were chosen to provide a balanced mix of classification and regression

tasks. During preprocessing, we remove all missing values and exclude temporal features, which tree-based ensembles cannot handle natively. Detailed dataset characteristics—including the number of samples, feature types, and task categories—are summarized in Table 4. For all experiments, we took a train-test split of 75-25; in cases where a validation set was needed, we took an additional train-validation split of 90-10 (in this case, the train-validation-test split would be 67.5-7.5-25).

Table 4: Summary of the local experiment datasets. Feature counts exclude the response and describe the input columns before one-hot encoding. Numerical and categorical counts follow the data types used by the experiment loaders; integer-coded predictors treated numerically are counted as numerical.

| Dataset | Code | #Samples | #Numerical | #Categorical | # Total | Task |
|---|---|---|---|---|---|---|
| Abalone | `abalone` | 4177 | 7 | 1 | 8 | Regression |
| Auto MPG | `auto_mpg` | 392 | 7 | 0 | 7 | Regression |
| Bike Sharing | `bike` | 17379 | 12 | 0 | 12 | Regression |
| Boston Housing | `boston` | 506 | 12 | 0 | 12 | Regression |
| California Housing | `california` | 20640 | 8 | 0 | 8 | Regression |
| Concrete | `concrete` | 1030 | 8 | 0 | 8 | Regression |
| Diamonds | `diamonds` | 53940 | 6 | 3 | 9 | Regression |
| Friedman | `friedman_1` | 1000 | 10 | 0 | 10 | Regression |
| Student Performance | `student` | 649 | 15 | 17 | 32 | Regression |
| Wine Quality | `wq` | 4898 | 11 | 0 | 11 | Regression |
| Adult | `adult` | 45222 | 5 | 8 | 13 | Classification |
| Banknote Auth. | `banknote` | 1372 | 4 | 0 | 4 | Classification |
| Breast Cancer | `bc` | 569 | 30 | 0 | 30 | Classification |
| Bank Customer Churn | `churn` | 10000 | 8 | 2 | 10 | Classification |
| Diabetes | `diabetes` | 768 | 8 | 0 | 8 | Classification |
| MAGIC Gamma Telescope | `magic` | 19020 | 10 | 0 | 10 | Classification |
| Spambase | `spambase` | 4601 | 57 | 0 | 57 | Classification |
| Telco Churn | `telco` | 7032 | 4 | 15 | 19 | Classification |

We describe the setup used for the experiments in Sect. 6. All experiments are run with 10 repetitions each (for every method/dataset combination, with different data splitting and model seeds).

### D.1 Spectra and Generalization

To evaluate the general performance of SCATE, we use all datasets except `adult` and `california`. Because our theoretical analysis is limited to RFs, this experiment focuses exclusively on them. For each dataset, we performed the following steps:

- After splitting the data (train, validation, test) and fitting an RF to the training set, we constructed the RF kernel $\boldsymbol{K}$ and computed its exact eigendecomposition. The resulting eigenvalues were sorted to generate a decay curve. We also computed the cross-kernel matrices $\boldsymbol{K}_{train,test}$ and $\boldsymbol{K}_{train,validation}$.
- We evaluate varying numbers of learned eigenfunctions, $P$, using the sequence `[2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 175, 200]`. We capped $P$ at 200, as subsequent eigenfunctions correspond to very small eigenvalues and contain minimal information.
- **Oracle Baseline:** For each value of $P$, we calculate the Oracle baseline (Sect. 5) by computing its approximation of $\boldsymbol{K}_{train,test}$ and its resulting predictions on the test data.
- For each $P$, we train two separate SCATE models (Sect. 5) for 200 epochs to learn the eigenfunction mappings: a small model (`width=16`, `depth=2`) and a large model (`width=128`, `depth=4`).
- The test data is passed through each model to generate output eigenfunctions. These are used to compute the approximated kernel $\hat{\boldsymbol{K}}_{train_test}$, which was then multiplied by the training labels $y$ to produce predictions. Performance is measured using accuracy for classification and $R^2$ for regression.

The resulting visualizations from this process are presented in Fig. 2.

### D.2 Naïve Benchmark

In this experiment, we evaluate SCATE against two simple distillation baselines for RFs:

- **Naïve MLP:** This approach uses the identical network architecture as SCATE. However, instead of learning eigenfunctions, it is trained to directly predict the output of each tree in the base model. This direct distillation scheme is inspired by [40].
- **Naïve RF:** A naturally smaller RF created by restricting the `max_depth` and `n_estimators` hyperparameters. Because default `scikit-learn` objects store metadata beyond what is needed for inference, we extract only the essential matrices into a custom `minimal_RF` object to ensure an accurate model size comparison.

We then benchmark SCATE against the Naïve MLP, the Naïve RF, and the Base RF, on all datasets excluding `adult` and `california`, using the following pipeline:

- We split the data, train the baseline RF, and construct the kernel exactly as described in the visualization experiment. We additionally record the performance of the baseline RF, as a measure of the performance we expect to converge to for all models.
- Based on earlier visualization results (Fig. 2) showing that SCATE learns most effectively up to 50 eigenfunctions, we fix $P = 50$. Instead of varying $P$, we sweep the network architectures for SCATE and the Naïve MLP using combinations of `widths=[4, 8, 16, 32, 64, 128]` and `depths=[1, 2, 3, 4, 5]`.
- Concurrently, we evaluate the Naïve RF baselines across combinations of `n_estimators=[10, 50, 100, 200, 500]` and `max_depth=[3, 4, 5, 6, 7, 8, 9, None]`.
- We train the models on their respective targets (eigenfunctions for SCATE, tree outputs for the MLP, and raw data for the Naïve RF). We then record both the predictive performance and the final model size for every configuration.
- For each dataset and method, we isolate the best-performing configurations at different model sizes to construct a convex Pareto frontier. Because model sizes scale differently across methods, some curves may contain very few data points if a majority of the configurations fall below the optimal frontier.

The resulting Pareto curves are plotted in Fig. 3.

### D.3 Compression Benchmark

For the compression benchmark, we test how well different methods perform compression. We focus on SCATE and FP-SCATE (which is SCATE applied to a model already shrunk by ForestPrune) under two strict memory limits: 10KB and 100KB. Unlike the earlier experiments, we also include GBMs here to compare performance.

Given all datasets in the pipeline, our benchmark proceeds as follows:

- Similar to the earlier tests, we start by splitting the data, training a standard Random Forest, and creating the kernel matrix $\boldsymbol{K}$ for each dataset.
- We test a wide range of settings (hyperparameters) for each method to see how they balance model size against accuracy. For some models, this amounts to an explicit accuracy loss bound — for others, tuning occurs by varying some regularisation hyperparameter, which would implicitly affect the loss in accuracy. For every combination of settings and dataset, we record the compressed model's size and performance. Since different algorithms output different types of models, we measure size fairly by only counting size of the bare minimum objects needed to make predictions.
- For each method and dataset, we find the best-performing model that fits under the 10KB and 100KB limits. The score of this best model becomes our final result for that budget, and we summarize these results in Table 1.
- Finally, we repeat this exact same process for the GBM models, constructing a smoother matrix $\boldsymbol{S}$ instead of an RF kernel. The results for the GBM tests are shown in Table 2.

Each method ends up being run around $50$ times for each dataset and seed, and the runs are doubled, since we evaluate on both RFs and GBMs, resulting in $50 \times 18 \times 10 \times 2 = 15000$ runs for each

method, and we benchmark 7 methods in total, resulting in $\sim 10^5$ runs in total. To address these computational demands, we run the compression benchmark from a high performance computing partition. Other experiments were run from a Macbook Pro, with an M5 Pro chip with 18-core CPU and 20-core GPU, and 24GB RAM. In order to accurately compare training and inference time between methods, we conduct a small runtime experiment on this machine, which can be found in Appx. E.6

This section details the implementation of each method included in our compression benchmark. For all baseline methods, we outline any structural modifications required to adapt them to our experimental framework, alongside the hyperparameter grids searched to identify the optimal model for each dataset.

- **SCATE:** Following the setup used in our naïve benchmark, we fix $P = 50$ and evaluate configurations across `widths` $\in \{4, 8, 16, 32, 64, 128\}$ and `depths` $\in \{1, 2, 3, 4, 5\}$. For GBM compression, because our smoother matrix **S** is strictly defined for MSE loss, we train a regressor for binary classification tasks and interpret the continuous outputs as linear probabilities.
- **FP:** The ForestPrune algorithm [65] lacks native support for classification tasks. To address this, we apply the same workaround used in our own models by fitting a regressor for binary classification. We identify the optimal model by sweeping a hyperparameter grid generated via `np.flip(np.logspace(-2, 1.5, 50))`, mirroring the configuration in the algorithm's official implementation.
- **FP-SCATE:** This pipeline sequentially applies FP followed by SCATE. We first isolate the best-performing FP model using the hyperparameter sweep described above. We then compress that specific model using SCATE, evaluating it over the same hyperparameter grid as base SCATE.
- **LOP:** For the LOP method [33], we search for the best model by sweeping tolerance values continuously from 0.01 to 1. To manage computational overhead, we impose a strict 10-minute timeout per run. In practice, this limit was only triggered a handful of times on the most computationally demanding datasets.
- **FIRE and TE:** Similar to FP, neither the FIRE [66] nor TreeExtract [67] algorithms natively handle compression for classification. Consequently, we employ the same regressor workaround for binary classification tasks. For FIRE, we sweep the $\alpha$ hyperparameter across values from 10 to 100. For TE, we evaluate models across a $\lambda$ range defined by `np.flip(np.logspace(1, 4, 50))`.
- **SSF:** Because the SSF algorithm [2] is designed exclusively for compressing Random Forests, we exclude it from our GBM benchmarks. We determine the optimal SSF models by sweeping threshold values between 0 and 0.5, which covers the entire hyperparameter range specified in the original paper.

### D.4 SCATE Size Specification

**SCATE model size and architecture selection.** Let $d$ denote the input dimension after preprocessing, $w$ the hidden-layer width, $L \geq 1$ the number of hidden layers, and $m$ the number of spectral outputs. All hidden layers have width $w$. The network contains

$$P(d, w, L, m) = \underbrace{w(d+1)}_{\text{input layer}} + \underbrace{(L-1)w(w+1)}_{\text{hidden layers}} + \underbrace{m(w+1)}_{\text{output layer}} \tag{7}$$

parameters, including a bias for each hidden and output unit.

For the scalar regression and binary classification experiments, the prediction also uses a vector of $m$ spectral coefficients. The network parameters are stored as 32-bit floating-point tensors; the coefficient vector is a 64-bit floating-point NumPy array in the reported implementation. Thus, their numerical payload requires

$$B_{\text{payload}} = 4P(d, w, L, m) + 8m \qquad \text{bytes.} \tag{8}$$

The size recorded in our benchmark additionally includes implementation-dependent overhead. Specifically, it is measured as

$$B_{\text{measured}} = |\text{torch.save}(\text{state_dict}(\theta))| + \text{getsizeof}(\mathbf{c}), \tag{9}$$

where the first term is the number of bytes in the serialized network state dictionary and $\mathbf{c} \in \mathbb{R}^m$ is the coefficient array. Equivalently, for the data types above,

$$B_{\text{measured}} = 4P(d, w, L, m) + 8m + H_{\text{torch}} + H_{\text{array}},$$

where $H_{\text{torch}}$ is the state-dictionary serialization overhead and $H_{\text{array}}$ is the additional array-object overhead counted by `getsizeof`. These overheads depend on the software environment and representation.

Given a byte budget $B$, a user can enumerate candidate values of $(w, L, m)$, construct the corresponding network and coefficient array, and retain configurations satisfying $B_{\text{measured}} \leq B$. A configuration can then be selected among the feasible candidates using validation performance. The parameter-count formula provides an initial screening rule, while direct measurement accounts for serialization overhead.

# E Additional Results

We present additional results from the visualization experiment and the naïve benchmark, experiments comparing different methods of learning eigenfunctions, and runtime experiments.

## E.1 Evaluating Different Methods for Learning Eigenfunctions

We evaluate and compare the performance of two existing neural eigenfunction estimation approaches—NeuralEF [32] and NeuralSVD [82]—against our proposed method, SCATE. Our evaluation follows two experimental setups: first, a capacity-constrained setting similar to the naïve benchmark, where we fix the model size and vary the number of target eigenfunctions $P$; second, a budget-constrained setting similar to the compression benchmark, where we fix $P$ and vary model sizes to identify the optimal configuration. Due to the relatively high computational demands of NeuralEF and NeuralSVD, we restrict this comparative analysis to five small datasets: `auto_mpg`, `student`, `boston`, `bc`, and `diabetes`, serving as an illustrative side-by-side comparison. For SCATE, we maintain the experimental settings detailed in Sect. 5 and Appx. D.

Because the primary contributions of NeuralEF and NeuralSVD are novel learning objectives rather than specific model architectures, their objectives can theoretically be adapted to any neural network. However, to ensure a fair comparison, we utilize the network designs specified in each method's official implementation. We briefly summarize their methodologies below:

- **NeuralEF:** NeuralEF formulates kernel eigendecomposition as a series of asymmetric maximization problems. Because this formulation relies on the deflation of previously computed modes, the optimization exhibits a strict hierarchical dependency where latter eigenfunctions depend on the convergence of earlier ones. While their original objective is a constrained optimization problem, in practice, NeuralEF relaxes orthogonality into explicit penalty terms within the loss function and enforces unit-norm requirements via a $L^2$ batch normalization layer. This structure requires training a neural network for each of the $P$ eigenfunctions it aims to learn. Furthermore, because its objective is fundamentally rooted in maximizing the Rayleigh quotient, NeuralEF is strictly confined to self-adjoint operators. As a result, it cannot compute the singular value decomposition (SVD) of kernels, limiting its applicability in our context exclusively to Random Forest (RF) compression.
- **NeuralSVD:** In contrast, NeuralSVD frames spectral decomposition as a fully unconstrained optimization task based on Schmidt's low-rank approximation theorem. By introducing a "NestedLoRA" objective, it recovers the ordered singular functions by breaking the symmetry of the low-rank approximation. This approach provides several improvements over NeuralEF. First, NeuralSVD eliminates the need for explicit orthogonality penalties and normalization layers; instead, it learns unnormalized singular functions that naturally absorb the scale of their corresponding singular values. Second, rather than using a sequentially trained network, NeuralSVD employs a joint nesting technique that unifies the learning of all $P$ modes into a single global objective function. This allows the model to simultaneously and optimize all functions within a single, shared neural network, similar to SCATE. Finally, because its objective natively supports SVD, NeuralSVD is capable of compression of both RFs and Gradient Boosting Machines (GBMs).

Table 5: RF compression benchmark for the three eigenfunction learning methods, averaged over 10 trials. Best result per row in bold. NA indicates that the model could not reach the target size.

| | | Size = 10KB | | | Size = 100KB | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **RF** | **NeuralEF** | **NeuralSVD** | **SCATE** | **NeuralEF** | **NeuralSVD** | **SCATE** |
| auto_mpg | 0.69 | $0.086 \pm 0.011$ | $0.237 \pm 0.020$ | $\mathbf{0.453 \pm 0.012}$ | $0.192 \pm 0.005$ | $0.320 \pm 0.009$ | $\mathbf{0.497 \pm 0.009}$ |
| bc | 0.96 | $0.732 \pm 0.018$ | $0.804 \pm 0.028$ | $\mathbf{0.984 \pm 0.003}$ | $0.881 \pm 0.016$ | $0.949 \pm 0.008$ | $\mathbf{0.987 \pm 0.003}$ |
| boston | 0.85 | $0.149 \pm 0.011$ | $0.220 \pm 0.015$ | $\mathbf{0.775 \pm 0.013}$ | $0.149 \pm 0.011$ | $0.252 \pm 0.013$ | $\mathbf{0.814 \pm 0.012}$ |
| diabetes | 0.76 | $0.654 \pm 0.011$ | $0.658 \pm 0.007$ | $\mathbf{0.786 \pm 0.010}$ | $0.654 \pm 0.011$ | $0.660 \pm 0.007$ | $\mathbf{0.794 \pm 0.010}$ |
| student | 0.85 | NA | $0.116 \pm 0.013$ | $\mathbf{0.854 \pm 0.014}$ | $0.134 \pm 0.007$ | $0.337 \pm 0.018$ | $\mathbf{0.857 \pm 0.013}$ |
| **Average Rank** | | 3.00 | 2.00 | **1.00** | 3.00 | 2.00 | **1.00** |

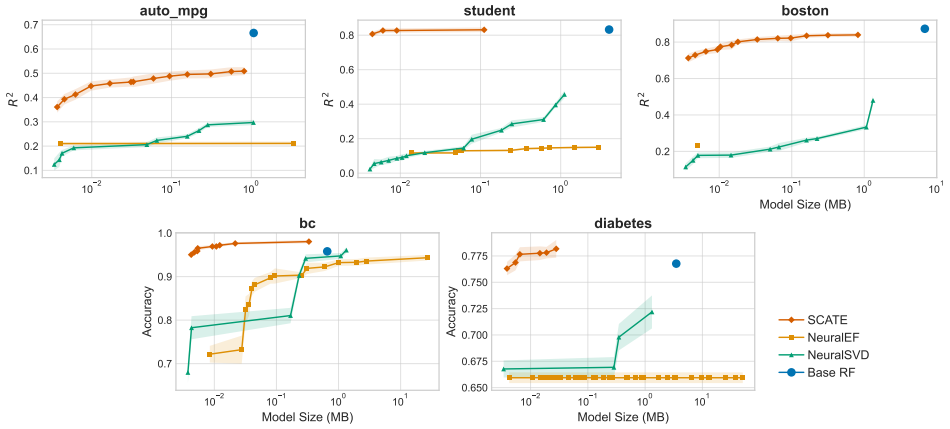



Figure 4: Rate-performance Pareto curves across five benchmarkd datasets for three eigenfunction learning methods. Shaded regions indicate standard errors.

From these results, we can see SCATE dominating compression performance. There are several reasons that may have caused subpar performance from the two competing methods, despite their theoretical complexity.

First, there is a fundamental difference in their optimization objectives: while NeuralEF and NeuralSVD are designed for mathematically exact spectral decomposition, SCATE is designed for kernel reconstruction. This can make a big difference in compression settings, where a network's learning capacity is heavily limited. In particular, both NeuralEF and NeuralSVD dedicate a substantial proportion of their capacity towards learning eigenvectors that are strictly orthogonal. This objective may be too complex for the limited learning capacity of small networks and is better suited for more expressive models.

In contrast, SCATE treats orthogonality as a soft penalty. Since top eigencomponents have strong signals, we do not necessarily need the orthogonality property to learn the top eigenfunctions effectively. Ultimately, this performance gap does not imply that NeuralEF or NeuralSVD are inferior methods; rather, their objectives and motivations are more appropriate for different settings than our own. As a consequence, we select SCATE as the primary method for learning the eigenfunctions in our RF compression experiments.

### E.2 Additional Visualization Results

Complementing the two-dataset visualization in Fig. 2, Fig. 5 (regression) and Fig. 6 (classification) show the eigenvalue decay, kernel approximation error, and predictive performance for the thirteen remaining datasets in the benchmark.

These findings further illustrate that substantial improvements in compression performance are closely associated with a steep decay in the kernel's eigenvalues. Moreover, SCATE consistently matches Oracle performance, demonstrating robustness even when the kernel approximation error stops improving due to noise in the trailing eigencomponents.

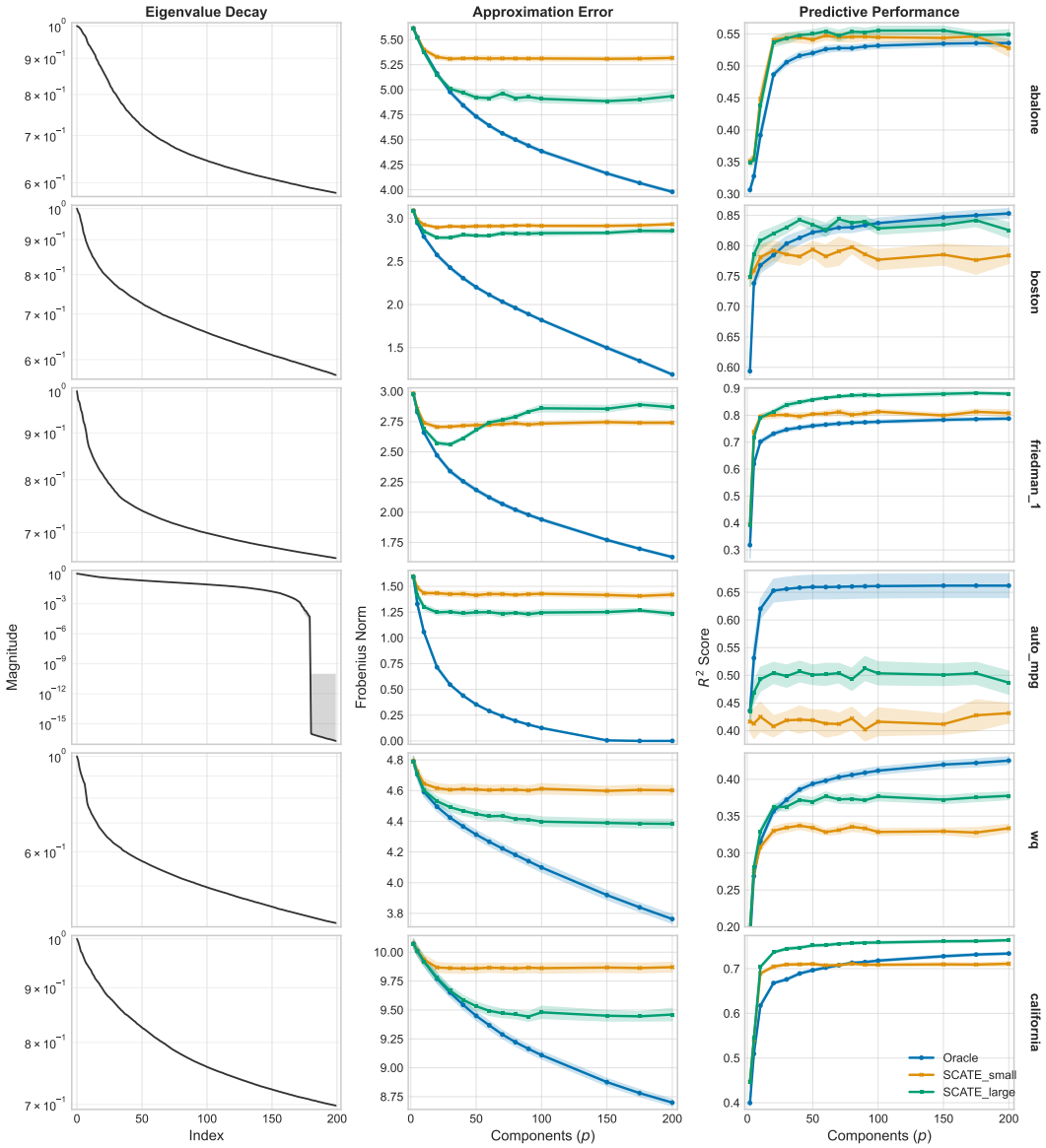



Figure 5: Spectral decay, kernel approximation error, and predictive performance across the remaining six regression datasets in the benchmark. Shaded regions indicate standard errors.

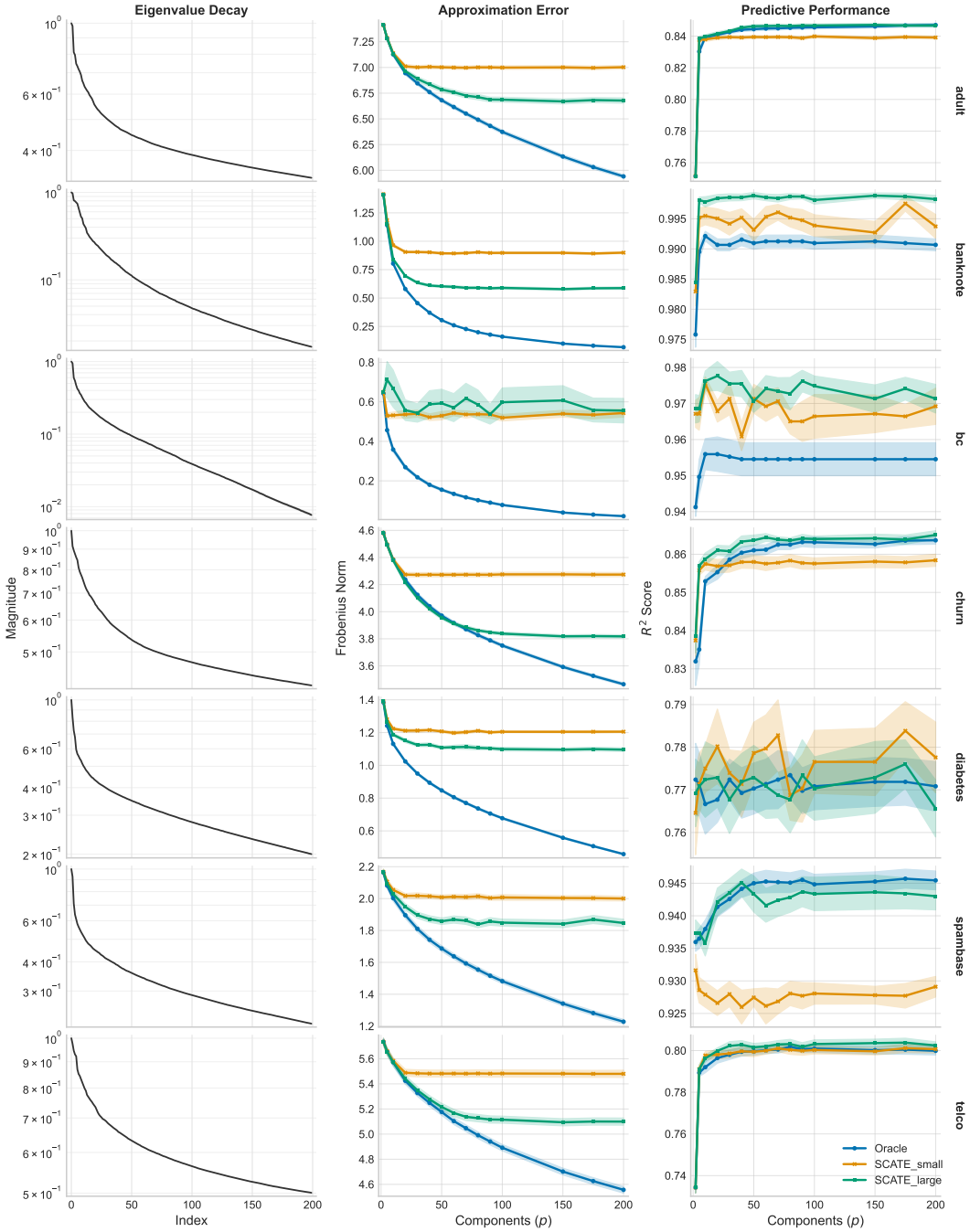



Figure 6: Spectral decay, kernel approximation error, and predictive performance across the remaining seven classification datasets in the benchmark. Shaded regions indicate standard errors.

### E.3 Additional Naïve Benchmark Results

In Figure 7, we present the Pareto frontiers for all remaining datasets evaluated in the naïve benchmark, supplementing the initial results shown in Figure 3. These plots demonstrate that SCATE consistently serves as the dominant method for RF compression compared to naïve alternatives.

While there are isolated instances where baseline methods achieve higher scores, these occurrences typically coincide with anomalous scenarios where the compressed models outperform the uncompressed Base RF. While these performance gains are possible, they are neither predictable nor broadly replicable, and thus cannot be relied on in practice. In contrast, SCATE maintains stable performance across diverse scenarios, validating its robustness as a highly effective methodology for RF compression.

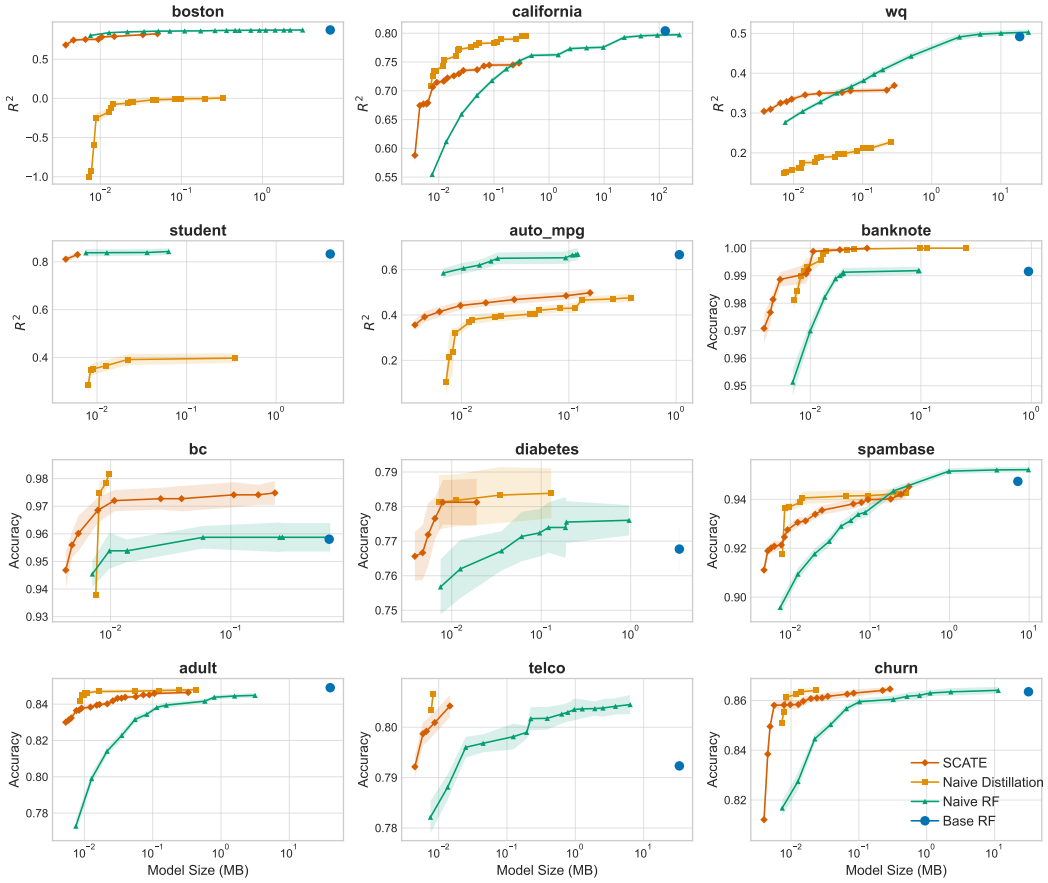



Figure 7: Rate-performance Pareto curves across the remaining twelve benchmark datasets. Shaded regions indicate standard errors.

### E.4 Comprehensive Compression Benchmark Results

As a supplement to Tables 1 and 2, we present the complete compression benchmark results in Tables 6 and 7. This expanded table reports the standard errors for all evaluated methods across each dataset. To identify the most competitive approaches, methods achieving performance within one standard error of the top-performing model are jointly highlighted in bold, though they remain ordered by their absolute mean performance. Notably, while the mean performance of several baseline models may initially appear comparable to our own, a more granular analysis incorporating these standard errors reveals that our method is superior, in the vast majority of scenarios, and validates our empirical rankings. We additionally report the size of the base RF and GBM models, which for 10KB show a compression factor around $10^3$ for RFs, and $10^2$ for GBMs, which is relatively inline with the values often tested for within the literature [33, 65]. To be fair with previous works, which reported leaf count as a measure of size, we strip down each base model to simply the matrices needed to perform inference (features, thresholds, left, right, and values) before computing its size.

Table 6: RF compression benchmark. Scores and base sizes are mean ± standard error across available runs. Base sizes use MiB. Bold entries lie within one standard error of the highest mean. Superscripts $[n]$ indicate fewer than ten contributing runs. NA denotes no within-budget result; TO denotes no candidate result within time limit.

| | | | Size = 10KB | | | | | | | Size = 100KB | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **RF** | **Base Size** | **ForestPrune** | **LOP** | **FIRE** | **TreeExtract** | **SSF** | **SCATE** | **FP-SCATE** | **ForestPrune** | **LOP** | **FIRE** | **TreeExtract** | **SSF** | **SCATE** | **FP-SCATE** |
| abalone | 0.548 | 31.180 ± 0.110 | 0.347 ± 0.010 | 0.504 ± 0.009 | 0.282 ± 0.016 | NA | NA | **0.561 ± 0.004** | 0.543 ± 0.010 | 0.533 ± 0.004 | 0.511 ± 0.010 | 0.283 ± 0.016 | 0.548 ± 0.004 | NA | **0.568 ± 0.003** | 0.545 ± 0.010 |
| auto_mpg | 0.874 | 4.864 ± 0.015 | 0.630 ± 0.025 | 0.813 ± 0.015 | 0.676 ± 0.015 | NA | NA | 0.852 ± 0.010 | **0.867 ± 0.008** | 0.820 ± 0.016 | 0.820 ± 0.014 | 0.677 ± 0.015 | 0.840 ± 0.014 | 0.812 ± 0.010 | 0.869 ± 0.008 | **0.878 ± 0.008** |
| bike | 0.94 | 151.507 ± 0.311 | 0.434 ± 0.025 | NA | TO | TO | **0.699 ± 0.001** | 0.678 ± 0.007 | 0.684 ± 0.004 | 0.721 ± 0.018 | NA | TO | TO | 0.699 ± 0.001 | **0.855 ± 0.003** | 0.843 ± 0.003 |
| boston | 0.855 | 6.867 ± 0.020 | 0.600 ± 0.016 | **0.805 ± 0.023** | 0.684 ± 0.016 | NA | NA | 0.775 ± 0.013 | **0.790 ± 0.012** | 0.800 ± 0.015 | 0.818 ± 0.025 | 0.690 ± 0.016 | 0.810 ± 0.014 | 0.717 ± 0.013 | 0.814 ± 0.012 | **0.833 ± 0.011** |
| california | 0.804 | 129.371 ± 0.309 | 0.449 ± 0.027[9] | 0.684 ± 0.006 | TO | TO | NA | 0.700 ± 0.003 | **0.707 ± 0.006**[4] | 0.676 ± 0.010 | **0.744 ± 0.005** | TO | TO | NA | **0.746 ± 0.004** | **0.743 ± 0.004**[4] |
| concrete | 0.907 | 14.422 ± 0.025 | 0.504 ± 0.015 | 0.741 ± 0.021 | 0.730 ± 0.014 | NA | NA | 0.801 ± 0.008 | **0.820 ± 0.005** | 0.758 ± 0.014 | 0.855 ± 0.005 | 0.746 ± 0.014 | 0.894 ± 0.003 | **0.901 ± 0.004** | 0.862 ± 0.003 | 0.861 ± 0.003 |
| diamonds | 0.981 | 200.838 ± 0.774 | 0.778 ± 0.053[4] | NA | TO | TO | NA | **0.937 ± 0.005** | 0.594 ± 0.141[4] | 0.932 ± 0.003 | NA | TO | TO | 0.930 ± 0.000 | **0.959 ± 0.001** | 0.599 ± 0.142[4] |
| friedman_1 | 0.851 | 14.328 ± 0.016 | 0.474 ± 0.023 | 0.759 ± 0.012 | 0.482 ± 0.025 | NA | NA | **0.822 ± 0.006** | 0.811 ± 0.009 | 0.736 ± 0.014 | 0.825 ± 0.013 | 0.494 ± 0.024 | 0.794 ± 0.007 | NA | **0.863 ± 0.004** | 0.855 ± 0.006 |
| student | 0.851 | 4.077 ± 0.014 | 0.644 ± 0.030 | 0.804 ± 0.015 | 0.710 ± 0.013 | NA | 0.818 ± 0.015 | **0.854 ± 0.014** | 0.836 ± 0.019 | 0.838 ± 0.013 | 0.805 ± 0.015 | 0.716 ± 0.013 | 0.791 ± 0.011 | 0.818 ± 0.015 | **0.857 ± 0.013** | 0.840 ± 0.019 |
| wq | 0.496 | 18.393 ± 0.070 | 0.236 ± 0.012 | 0.312 ± 0.006 | 0.307 ± 0.010 | NA | NA | **0.336 ± 0.008** | **0.332 ± 0.006** | 0.350 ± 0.010 | **0.358 ± 0.008** | 0.340 ± 0.008 | 0.281 ± 0.005 | 0.243 ± 0.010 | **0.362 ± 0.007** | **0.360 ± 0.007** |
| adult | 0.851 | 39.593 ± 0.200 | 0.792 ± 0.003 | 0.827 ± 0.003 | TO | TO | NA | **0.839 ± 0.001** | 0.834 ± 0.001[6] | 0.837 ± 0.001 | 0.835 ± 0.001 | TO | TO | **0.848 ± 0.001** | **0.847 ± 0.001** | 0.846 ± 0.001[6] |
| banknote | 0.993 | 0.964 ± 0.010 | 0.874 ± 0.009 | 0.986 ± 0.004 | 0.992 ± 0.002 | NA | 0.999 ± 0.001 | **1.000 ± 0.000** | **1.000 ± 0.000** | 0.919 ± 0.011 | 0.986 ± 0.004 | 0.992 ± 0.002 | 0.892 ± 0.003 | **1.000 ± 0.000** | **1.000 ± 0.000** | **1.000 ± 0.000** |
| bc | 0.962 | 0.677 ± 0.014 | 0.950 ± 0.007 | 0.922 ± 0.027 | 0.955 ± 0.005 | NA | 0.976 ± 0.004 | **0.984 ± 0.003** | 0.980 ± 0.003 | 0.951 ± 0.007 | 0.922 ± 0.027 | 0.955 ± 0.005 | 0.920 ± 0.007 | 0.980 ± 0.003 | **0.987 ± 0.003** | **0.985 ± 0.003** |
| churn | 0.862 | 31.671 ± 0.177 | 0.823 ± 0.004 | 0.844 ± 0.006 | 0.850 ± 0.002 | NA | NA | **0.859 ± 0.002** | 0.852 ± 0.007 | 0.853 ± 0.002 | 0.850 ± 0.004 | 0.850 ± 0.002 | 0.851 ± 0.002 | 0.828 ± 0.004[3] | **0.863 ± 0.002** | 0.855 ± 0.008 |
| diabetes | 0.759 | 3.493 ± 0.040 | 0.757 ± 0.008 | 0.741 ± 0.014 | 0.765 ± 0.009 | NA | NA | **0.786 ± 0.010** | **0.794 ± 0.009** | 0.764 ± 0.009 | 0.741 ± 0.014 | 0.765 ± 0.009 | 0.661 ± 0.010 | 0.733 ± 0.007 | **0.794 ± 0.010** | **0.801 ± 0.009** |
| magic | 0.88 | 27.142 ± 0.120 | 0.793 ± 0.006 | 0.818 ± 0.004 | **0.860 ± 0.001** | NA | NA | **0.860 ± 0.002** | **0.860 ± 0.002** | 0.846 ± 0.002 | 0.859 ± 0.002 | 0.868 ± 0.001 | 0.841 ± 0.001[9] | NA | **0.872 ± 0.001** | 0.869 ± 0.001 |
| spambase | 0.949 | 7.219 ± 0.037 | 0.868 ± 0.005 | 0.915 ± 0.003 | **0.937 ± 0.002** | NA | NA | 0.929 ± 0.001 | 0.926 ± 0.002 | 0.913 ± 0.002 | 0.933 ± 0.002 | **0.944 ± 0.002** | 0.894 ± 0.003 | **0.944 ± 0.002** | 0.939 ± 0.001 | 0.936 ± 0.001 |
| telco | 0.789 | 33.395 ± 0.142 | 0.747 ± 0.007 | 0.773 ± 0.008 | 0.789 ± 0.003 | NA | NA | **0.804 ± 0.003** | 0.800 ± 0.002 | 0.797 ± 0.002 | 0.775 ± 0.008 | 0.789 ± 0.003 | 0.789 ± 0.002 | 0.778 ± 0.004 | **0.808 ± 0.003** | 0.804 ± 0.002 |
| **Average Rank** | | | 4.67 | 3.86 | 4.08 | 6.44 | 5.50 | **1.53** | 1.92 | 4.22 | 4.72 | 5.31 | 5.19 | 4.67 | **1.56** | 2.33 |

Table 7: GBM compression benchmark. Scores and base sizes are mean ± standard error across available runs. Base sizes use MiB. Bold entries lie within one standard error of the highest mean. Superscripts $[n]$ indicate fewer than ten contributing runs. NA denotes no within-budget result; TO denotes no candidate result within time limit.

| | | | Size = 10KB | | | | | | Size = 100KB | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **GBM** | **Base Size** | **FIRE** | **ForestPrune** | **FP-SCATE** | **LOP** | **SCATE** | **TreeExtract** | **FIRE** | **ForestPrune** | **FP-SCATE** | **LOP** | **SCATE** | **TreeExtract** |
| abalone | 0.529 | 0.690 ± 0.004 | 0.465 ± 0.009 | 0.445 ± 0.004 | 0.547 ± 0.007 | 0.505 ± 0.006 | **0.562 ± 0.002** | 0.540 ± 0.003 | 0.465 ± 0.009 | 0.528 ± 0.004 | 0.552 ± 0.008 | 0.520 ± 0.007 | **0.571 ± 0.002** | 0.547 ± 0.004 |
| auto_mpg | 0.858 | 0.585 ± 0.004 | 0.820 ± 0.012 | 0.799 ± 0.016 | 0.835 ± 0.016 | 0.820 ± 0.011 | **0.852 ± 0.008** | **0.854 ± 0.008** | 0.820 ± 0.012 | 0.861 ± 0.011 | 0.847 ± 0.018 | 0.826 ± 0.010 | **0.872 ± 0.007** | 0.854 ± 0.008 |
| bike | 0.95 | 0.850 ± 0.002 | **0.880 ± 0.003** | 0.571 ± 0.010 | 0.624 ± 0.009 | NA | 0.618 ± 0.007 | TO | 0.880 ± 0.003 | 0.922 ± 0.001 | 0.868 ± 0.010 | **0.944 ± 0.001** | 0.919 ± 0.001 | TO |
| boston | 0.862 | 0.581 ± 0.005 | 0.777 ± 0.027 | 0.775 ± 0.026 | 0.710 ± 0.067 | **0.807 ± 0.026** | **0.790 ± 0.012** | **0.786 ± 0.015** | 0.777 ± 0.027 | **0.866 ± 0.014** | 0.757 ± 0.069 | 0.826 ± 0.026 | 0.842 ± 0.009 | 0.817 ± 0.013 |
| california | 0.829 | 0.803 ± 0.004 | 0.750 ± 0.004 | 0.583 ± 0.004 | 0.704 ± 0.003 | 0.758 ± 0.006 | 0.708 ± 0.004 | **0.792 ± 0.003** | 0.750 ± 0.004 | 0.800 ± 0.003 | 0.763 ± 0.006 | **0.823 ± 0.002** | 0.792 ± 0.002 | 0.792 ± 0.003 |
| concrete | 0.921 | 0.687 ± 0.005 | 0.845 ± 0.007 | 0.711 ± 0.008 | 0.814 ± 0.006 | 0.834 ± 0.007 | 0.789 ± 0.013 | **0.870 ± 0.004** | 0.845 ± 0.007 | 0.901 ± 0.004 | 0.868 ± 0.005 | 0.908 ± 0.003 | 0.879 ± 0.005 | **0.915 ± 0.003** |
| diamonds | 0.981 | 0.835 ± 0.001 | **0.939 ± 0.001** | 0.878 ± 0.003 | 0.935 ± 0.009 | TO | **0.942 ± 0.005** | NA | 0.939 ± 0.001 | 0.975 ± 0.000 | 0.960 ± 0.003 | TO | 0.970 ± 0.001 | **0.981 ± 0.000**[6] |
| friedman_1 | 0.910 | 0.717 ± 0.005 | 0.790 ± 0.010 | 0.680 ± 0.015 | 0.867 ± 0.016 | 0.874 ± 0.017 | 0.893 ± 0.007 | **0.917 ± 0.003** | 0.790 ± 0.010 | 0.899 ± 0.005 | 0.926 ± 0.015 | 0.905 ± 0.004 | **0.972 ± 0.001** | 0.917 ± 0.003 |
| student | 0.832 | 0.689 ± 0.006 | 0.752 ± 0.031 | 0.811 ± 0.014 | 0.699 ± 0.041 | 0.789 ± 0.033 | **0.837 ± 0.012** | 0.785 ± 0.014 | 0.752 ± 0.031 | **0.857 ± 0.009** | 0.701 ± 0.041 | 0.789 ± 0.033 | 0.838 ± 0.012 | 0.785 ± 0.014 |
| wq | 0.454 | 0.683 ± 0.005 | **0.346 ± 0.007** | 0.287 ± 0.005 | 0.331 ± 0.007 | 0.317 ± 0.007 | 0.333 ± 0.007 | 0.300 ± 0.006 | 0.346 ± 0.007 | **0.385 ± 0.007** | 0.356 ± 0.007 | **0.381 ± 0.006** | 0.358 ± 0.008 | 0.300 ± 0.006 |
| adult | 0.853 | 0.705 ± 0.003 | 0.814 ± 0.002 | 0.819 ± 0.002 | 0.841 ± 0.001 | 0.844 ± 0.002 | 0.846 ± 0.001 | **0.853 ± 0.001** | 0.814 ± 0.002 | 0.853 ± 0.001 | 0.849 ± 0.001 | 0.852 ± 0.001 | 0.853 ± 0.001 | **0.856 ± 0.001** |
| banknote | 0.985 | 0.471 ± 0.023 | 0.982 ± 0.002 | 0.947 ± 0.008 | 0.998 ± 0.002 | 0.977 ± 0.005 | **1.000 ± 0.000** | 0.991 ± 0.002 | 0.982 ± 0.002 | 0.986 ± 0.002 | 0.998 ± 0.002 | 0.977 ± 0.005 | **1.000 ± 0.000** | 0.995 ± 0.001 |
| bc | 0.938 | 0.310 ± 0.034 | 0.946 ± 0.004 | 0.936 ± 0.008 | 0.947 ± 0.036 | 0.937 ± 0.007 | **0.984 ± 0.003** | 0.945 ± 0.006 | 0.946 ± 0.004 | 0.949 ± 0.005 | 0.949 ± 0.036 | 0.937 ± 0.007 | **0.990 ± 0.002** | 0.952 ± 0.005 |
| churn | 0.861 | 0.711 ± 0.004 | 0.852 ± 0.002 | 0.847 ± 0.002 | 0.853 ± 0.006 | 0.862 ± 0.002 | 0.860 ± 0.002 | **0.864 ± 0.002** | 0.852 ± 0.002 | **0.864 ± 0.002** | 0.855 ± 0.006 | 0.862 ± 0.002 | **0.865 ± 0.002** | **0.866 ± 0.002** |
| diabetes | 0.744 | 0.592 ± 0.004 | 0.744 ± 0.011 | 0.768 ± 0.009 | 0.728 ± 0.026 | 0.755 ± 0.012 | **0.791 ± 0.011** | 0.777 ± 0.008 | 0.744 ± 0.011 | 0.779 ± 0.009 | 0.730 ± 0.027 | 0.755 ± 0.012 | **0.797 ± 0.010** | 0.783 ± 0.009 |
| magic | 0.88 | 0.731 ± 0.003 | 0.851 ± 0.002 | 0.811 ± 0.002 | 0.862 ± 0.002 | 0.854 ± 0.002 | 0.862 ± 0.001 | **0.866 ± 0.001** | 0.851 ± 0.002 | 0.873 ± 0.002 | 0.874 ± 0.002 | 0.871 ± 0.003 | 0.875 ± 0.001 | **0.881 ± 0.002** |
| spambase | 0.945 | 0.507 ± 0.003 | 0.916 ± 0.002 | 0.904 ± 0.002 | 0.926 ± 0.002 | 0.922 ± 0.003 | 0.928 ± 0.002 | **0.944 ± 0.001** | 0.916 ± 0.002 | 0.946 ± 0.001 | 0.935 ± 0.001 | 0.934 ± 0.003 | 0.938 ± 0.001 | **0.951 ± 0.001** |
| telco | 0.794 | 0.691 ± 0.007 | 0.793 ± 0.003 | 0.791 ± 0.003 | 0.793 ± 0.009 | 0.794 ± 0.004 | **0.806 ± 0.003** | **0.804 ± 0.003** | 0.793 ± 0.003 | 0.803 ± 0.003 | 0.798 ± 0.009 | 0.794 ± 0.004 | **0.809 ± 0.003** | 0.805 ± 0.003 |
| **Average Rank** | | | 3.83 | 5.28 | 3.67 | 3.67 | **2.06** | 2.50 | 5.50 | 2.64 | 4.19 | 3.94 | **2.06** | 2.67 |

### E.5 Visualizing Compression Frontiers

While Tables 6 and 7 provides a granular analysis of performance at discrete memory constraints, we further evaluate SCATE across a continuous capacity spectrum ranging from 1KB to 100KB. Plotting traditional Pareto frontiers for all individual methods often results in severe visual clutter, as disparate hyperparameter configurations across techniques lead to irregular and non-uniform model sizes. Instead, to ensure a clear comparison, we isntead distill these results into two curves each dataset. The first curve tracks the peak performance of SCATE evaluated at 1KB intervals. The second curve is a composition of the state-of-the-art pruning methods: at each 1KB threshold, it adopts the maximum score achieved by *any* competing baseline method (excluding the generally weaker FP-SCATE variant). This baseline then represents a best-in-class curve against which we can evaluate SCATE.




Figure 8: Visualization curves for the RF compression benchmark. The solid line represents SCATE, and the dashed line represents the composite method.

The resulting performance curves are illustrated in Figure 8 for RFs and Figure 9 for GBMs. An examination of these figures reveals SCATE's robustness, especially around the 10KB point. In Fig. 8, SCATE frequently beats the composite curve or performs highly competitively, with the




Figure 9: Visualization curves for the GBM compression benchmark. The solid line represents SCATE, and the dashed line represents the composite method.

exception of the `auto_mpg` dataset. In the GBM domain, the results are more tempered; SCATE slightly underperforms, though the differences are generally narrow. As discussed in Sect. 7, this dynamic occurs because SCATE's compression footprint is bounded by the complexity of the kernel rather than the size of the original model. Consequently, we can attribute these results to competing methods inherently struggling more to compress the larger parameter counts of RFs.

### E.6 Runtime Experiments

We evaluate the computational efficiency of our method and others through two experiments. First, we measure benchmark both training and inference times across three representative datasets. Then, we evaluate the trade-off between compression time and predictive performance under some fixed memory budgets. The second experiment is intended to give a more realistic view on the cost of searching for configurations, as opposed to just the runtime of the selected model.

**Training and Inference Benchmark** We assess the runtime of our experiments on three small-to-medium datasets: `student`, `wq`, and `telco`. We deliberately exclude the two largest datasets (`adult`

and `california`) from this specific analysis; the prohibitive computational scaling of baseline rule extractors, specifically FIRE [66] and TreeExtract [67], resulted in frequent timeouts on these datasets, preventing a complete comparison.

Because these compression methods are primarily motivated by strict deployment constraints, runtime is most meaningfully analyzed as a function of the final model's memory footprint - as a result, we evaluate runtime across a continuous capacity spectrum. We discretize the target memory footprint into 5KB intervals, between 0KB to 100KB. For each interval, we compute the average training and inference times across all hyperparameter configurations that yield a compressed model falling within that specific 5KB memory bucket. This procedure is conducted independently for both RF and GBM base models, generating independent training and inference time profiles for each dataset (resulting in 12 comparative plots in total). The resulting plots are produced in

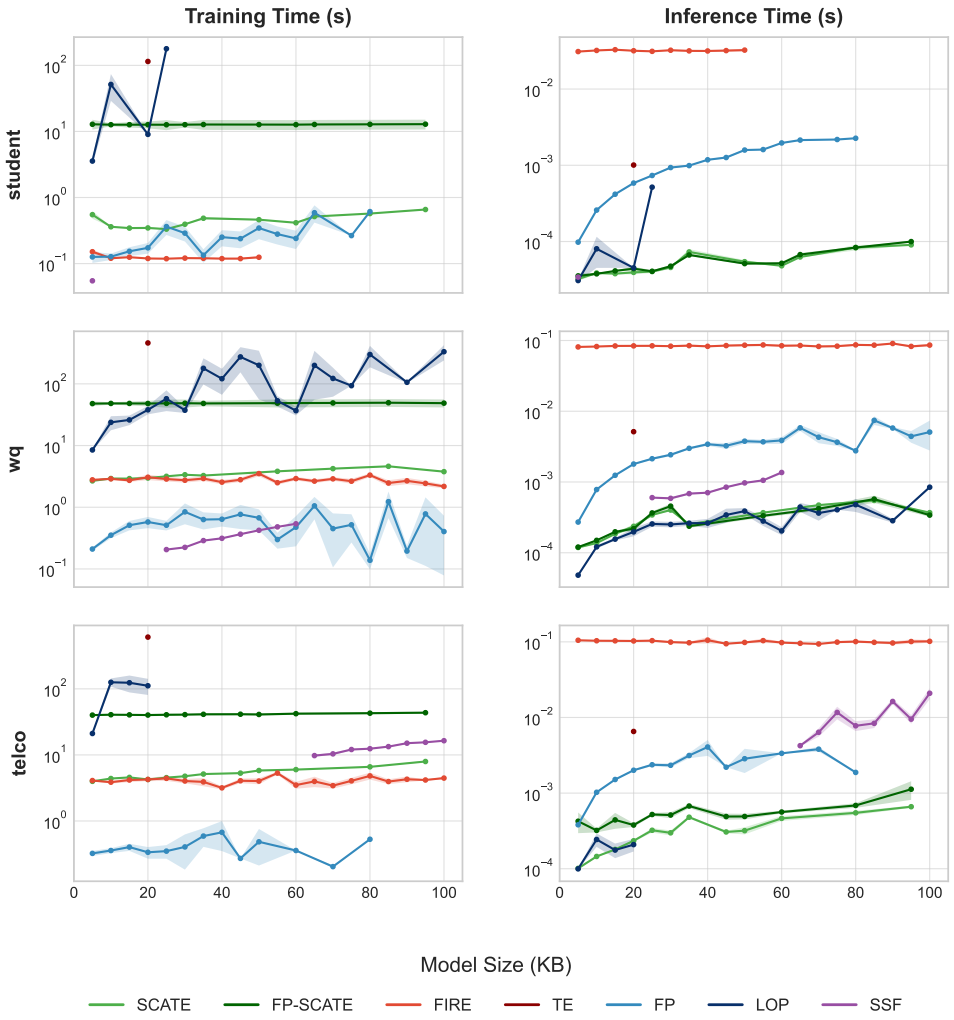



Figure 10: Training and inference time against model size plots for RFs, over three datasets

Interestingly, our evaluations indicate that runtime remains largely invariant across different model complexities for the majority of the evaluated algorithms. Therefore, an additional comprehensive summary of these computational costs, reported as means and standard errors, is provided in Tab. 8.

SCATE has the lowest mean inference time, with FP-SCATE also remaining competitive. This is relevant for constrained deployment scenarios, though it is possible that competing tree methods can be made more efficient by making greater use of tree parallelization, in the RF case. However, these measurements concern individual configurations - the cost of searching for an optimal configuration on the validation set is often a concern in practice, which we also explore below.

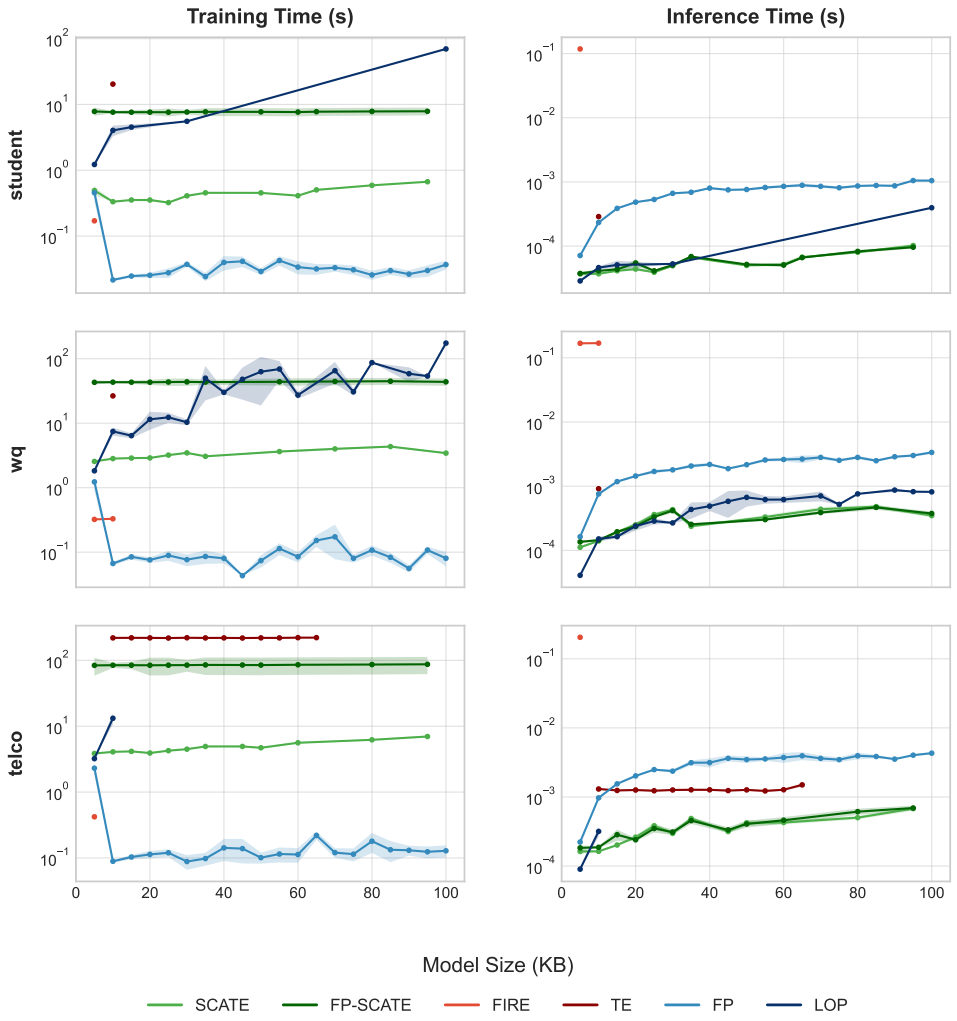



Figure 11: Training and inference time against model size plots for GBMs, over three datasets

Table 8: Average training and inference runtimes over the entire evaluated hyperparameter set. All values are reported in seconds as mean ± standard error. The fastest methods per column (including those within 1 standard error of the best) are bolded. SSF results in GBM are blank, as SSF is an RF-exclusive method.

| | Random Forest | | GBM | |
|---|---|---|---|---|
| **Method** | **Avg Training (s)** | **Avg Inference (s)** | **Avg Training (s)** | **Avg Inference (s)** |
| **SCATE** | $3.0402 \pm 0.3668$ | $\mathbf{2.32 \times 10^{-4} \pm 3.04 \times 10^{-5}}$ | $2.8506 \pm 0.3340$ | $\mathbf{2.37 \times 10^{-4} \pm 2.96 \times 10^{-5}}$ |
| **FP-SCATE** | $33.7156 \pm 2.6515$ | $3.06 \times 10^{-4} \pm 4.26 \times 10^{-5}$ | $45.5539 \pm 5.5014$ | $\mathbf{2.41 \times 10^{-4} \pm 2.99 \times 10^{-5}}$ |
| **LOP** | $108.8504 \pm 18.2936$ | $2.69 \times 10^{-4} \pm 3.49 \times 10^{-5}$ | $36.3631 \pm 7.8968$ | $4.01 \times 10^{-4} \pm 5.67 \times 10^{-5}$ |
| **FP** | $\mathbf{0.4319 \pm 0.0347}$ | $2.46 \times 10^{-3} \pm 2.34 \times 10^{-4}$ | $\mathbf{0.1445 \pm 0.0422}$ | $1.97 \times 10^{-3} \pm 1.61 \times 10^{-4}$ |
| **TE** | $393.9479 \pm 146.0017$ | $4.23 \times 10^{-3} \pm 1.66 \times 10^{-3}$ | $191.7248 \pm 19.0575$ | $1.18 \times 10^{-3} \pm 7.56 \times 10^{-5}$ |
| **SSF** | $6.3441 \pm 1.6434$ | $5.41 \times 10^{-3} \pm 1.52 \times 10^{-3}$ | – | – |
| **FIRE** | $2.7730 \pm 0.2128$ | $8.02 \times 10^{-2} \pm 3.61 \times 10^{-3}$ | $0.3109 \pm 0.0519$ | $1.65 \times 10^{-1} \pm 1.78 \times 10^{-2}$ |

Table 9: Additional RF runtime results on Churn and California Housing. Training and inference times are reported in seconds. NA indicates that SSF did not reach the 100 KB size threshold; TO indicates the 24-hour timeout. For SSF on Churn, only one size interval contributed.

| | Churn | | California | |
|---|---|---|---|---|
| **Method** | **Avg Training (s)** | **Avg Inference (s)** | **Avg Training (s)** | **Avg Inference (s)** |
| SSF | $27.5816 \pm 0.0000$ | $1.23\times10^{-1} \pm 0$ | NA | NA |
| FIRE | $5.8907 \pm 0.1828$ | $1.24\times10^{-1} \pm 7.03\times10^{-4}$ | TO | TO |
| FP | $2.0442 \pm 0.3313$ | $9.00\times10^{-3} \pm 1.04\times10^{-3}$ | $7.5740 \pm 1.0152$ | $2.68\times10^{-2} \pm 2.62\times10^{-3}$ |
| SCATE | $8.0607 \pm 0.3809$ | $4.66\times10^{-4} \pm 6.05\times10^{-5}$ | $11.1591 \pm 0.5400$ | $8.85\times10^{-4} \pm 1.04\times10^{-4}$ |

Table 10: Additional GBM runtime results on Churn and California Housing. Training and inference times are reported in seconds. For FIRE on Churn, only one size interval contributed, so the reported zero standard error does not quantify uncertainty. SSF is omitted because it is an RF-only method.

| | Churn | | California | |
|---|---|---|---|---|
| **Method** | **Training (s)** | **Inference (s)** | **Training (s)** | **Inference (s)** |
| FIRE | $0.5860 \pm 0.0000$ | $2.68\times10^{-1} \pm 0$ | $1.4372 \pm 0.0158$ | $5.14\times10^{-1} \pm 6.74\times10^{-3}$ |
| FP | $0.6671 \pm 0.2921$ | $5.46\times10^{-3} \pm 4.63\times10^{-4}$ | $1.5546 \pm 0.6215$ | $1.28\times10^{-2} \pm 1.10\times10^{-3}$ |
| SCATE | $16.5309 \pm 0.9629$ | $2.31\times10^{-3} \pm 2.17\times10^{-4}$ | $9.8198 \pm 0.6122$ | $6.54\times10^{-4} \pm 6.71\times10^{-5}$ |

**Additional runtime experiments** We additionally examine training and inference on the `churn` and `california` datasets for the faster SSF, FIRE, FP, and SCATE methods, and report the results in Tables 9 and 10. Values are means and standard errors across occupied model-size intervals in the nominal 5–100 KB range.

**Compression-configuration Benchmark** To compare runtime and predictive performance more jointly, we conduct an additional benchmark on eight datasets: `student`, `diabetes`, `spambase`, `wq`, `telco`, `california`, `magic`, and `diamonds`. Experiments use AMD EPYC 7702 CPUs on cluster computing, with five allocated CPU cores each seed and GPUs disabled. For each method, dataset, and ensemble type, we evaluate up to ten pre-specified configurations over up to ten data-split and model seeds. Each seed has a three-hour time limit. If the first two seeds both time out without completing a configuration, the remaining sessions for that combination are considered to have timed out as well.

For each completed seed, the reported compression-search time is the sum of shared preprocessing and training times of the evaluated configurations. For SCATE and FP-SCATE, this involves the construction and decomposition of the corresponding smoother matrices (with an additional FP step for FP-SCATE). A seed that reaches the time limit after completing at least one configuration is assigned a runtime of three hours, and its completed configurations are kept. Failed seeds and seeds with no completed configurations are marked as timeouts.

For each memory budget (10kB and 100kB), we select the configuration with the highest test score, and scale it by the uncompressed model's test score. The resulting ratio is a relative summary, since regression and classification problems are combined. Selecting models this way describes the best observed result within the search, as opposed to the performance of a configuration selected independently with validation data. We first average the normalized scores and search times across available seeds for each dataset. Figure 12 then reports the median and interquartile range across datasets with an eligible result.

For RF compression, SCATE attains a high median normalized score, at 0.976 and 0.987 for the 10 and 100KB budgets, respectively. It lies on the frontier of the displayed method medians at both budgets, where no stronger method is also faster (and in general it is also the strongest method). For GBMs, the trade-off differs. While SCATE remains competitive in terms of relative strength, the FIRE and FP methods are similarly powerful but provide an advantage in terms of runtime. This reflects our earlier results and claims, where the smaller GBM model base size allows some methods to be competitive with SCATE, but not for RF, where we generally dominate.

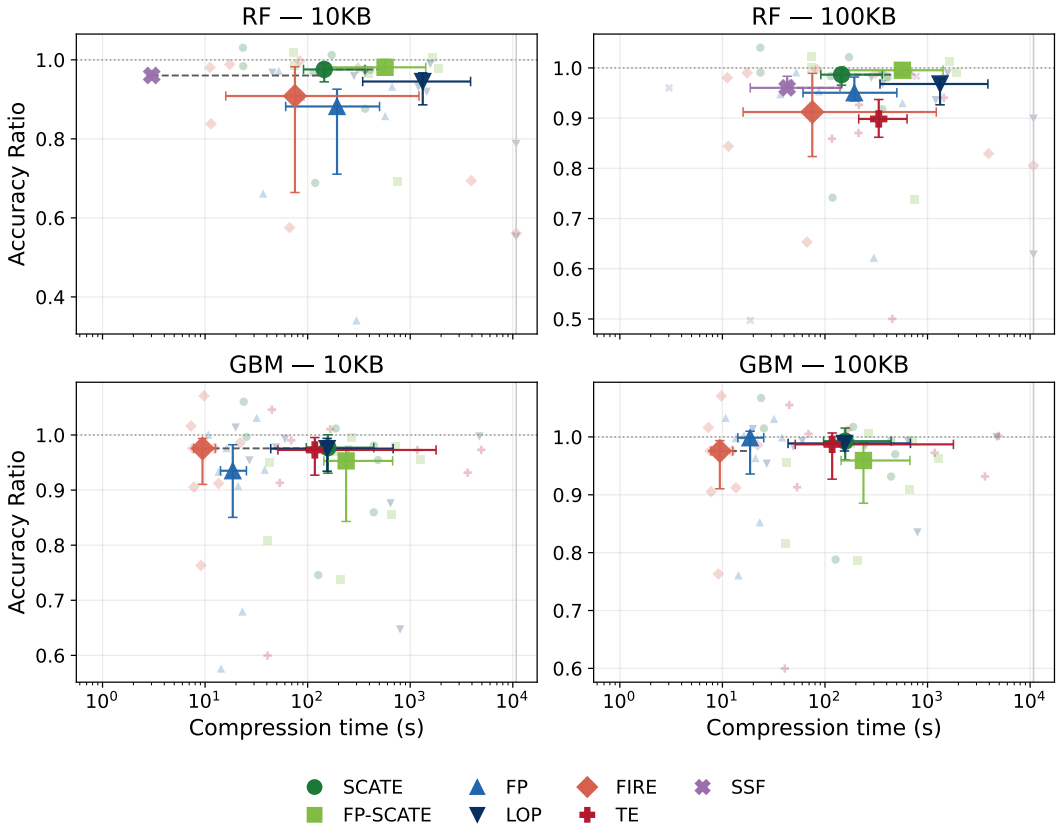



Figure 12: Compression-search time and normalized predictive performance for RF and GBM ensembles. Large markers show medians across datasets, after averaging over available seeds within each dataset; error bars show interquartile ranges, and faint markers show dataset-level means. The horizontal dotted line indicates base model performance, and the vertical line indicates the time limit.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract claims that we derive minimax-optimal convergence for RF regression under certain regularity conditions, which we define formally in Sect. 4, and prove in Appx. A. We also claim to derive a compression method for RFs and GBMs (Sect. 5), and that it performs competitively on this task (Sect. 6, and Appx. B).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations at length in Sect. 7, as well as in the appendix, whenever it is relevant.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Each theorem and claim made in the main text (primarily Sect. 4) is proven in Appx. A. The assumptions that we make for these theorems are clearly stated, and some of them are evaluated in Appx. E. Theorems that the proof relies upon are referenced whenever appropriate.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We include the model, code and data required to run our experiments within our supplemental material, as well as any necessary specifications to run them.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code, model and data is available via the supplementary material, and upon conclusion of the review process we intend to release the code publicly to facilitate further research.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup is summarized in the main text, and all substantial experimental design details are given in Appx. D, with results in Appx. E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are provided for all our experiments, and they were all run with repetition.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify some the computer resources that we use in our experiments in Appx. D. However, given the nature of the experiment and inconsistency in the amount of compute available to us at any given time, we are unable to report a 'total' compute count. Some information relating to our compute setup is also hidden to protect anonymity; we intend to provide this information upon the paper's public release.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and conformed to it, in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our contribution is foundational, meaning that there are no direct social impacts of our work. However, our paper partly revolves around knowledge distillation, which may carry potential positive impacts via optimizing resource consumption.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our model makes use of publicly available data, models and packages, with a general purpose use case. Therefore, there is no high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: All external methods and datasets used in, or are relevant to, our paper are appropriately referenced. Any modifications on these methods and datasets are also stated.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Our asset contribution is a codebase containing our model, the scripts we use for our experiments, and the datasets used. General documentation is provided via both comments in the codebase and in the paper itself.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Our paper does involve research with any human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We did not use LLMs in the core method development stages of our paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.