Statistical Inference for Gradient Boosting Regression

Haimo Fang[1], Kevin Tan[2]*, Giles Hooker[2]
[1]School of Economics, Fudan University
[2]Department of Statistics and Data Science, The Wharton School, University of Pennsylvania

Abstract

Gradient boosting is widely popular due to its flexibility and predictive accuracy. However, statistical inference and uncertainty quantification for gradient boosting remain challenging and under-explored. We propose a unified framework for statistical inference in gradient boosting regression. Our framework integrates dropout or parallel training with a recently proposed regularization procedure called Boulevard that allows for a central limit theorem (CLT) for boosting. With these enhancements, we surprisingly find that *increasing* the dropout rate and the number of trees grown in parallel at each iteration substantially enhances signal recovery and overall performance. Our resulting algorithms enjoy similar CLTs, which we use to construct built-in confidence intervals, prediction intervals, and rigorous hypothesis tests for assessing variable importance in only $O(nd^2)$ time with the Nyström method. Numerical experiments verify the asymptotic normality and demonstrate that our algorithms perform well, do not require early stopping, interpolate between regularized boosting and random forests, and confirm the validity of their built-in statistical inference procedures.

1 Introduction

Gradient boosting (Friedman, 2000), particularly through widely used implementations such as XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), has become one of the most powerful and widely adopted methods for supervised learning, especially on tabular data. However, this flexibility and predictive accuracy come at a cost. Unlike their base learners – such as decision trees or linear models – boosting methods are typically far less interpretable, and uncertainty quantification for their predictions is far from straightforward. While point estimates may suffice for ad hoc prediction tasks, they overlook randomness inherent in both the data and the algorithm itself. Uncertainty quantification therefore asks a central question: *if a new dataset were collected and models retrained, how different would the resulting predictions be?* This is a central ingredient in propagating uncertainty through forecast models, prioritizing data collection and a host of other downstream tasks.

In this light, various methods have been proposed for uncertainty quantification in boosting. These include Langevin boosting (Tan et al., 2023; Ustimenko and Prokhorenkova, 2022; Malinin et al., 2021), k-nearest neighbors-based techniques (Brophy and Lowd, 2022), Gaussian graphical models (Chen and Wang, 2023), quantile regression approaches (Yin et al., 2023), and probabilistic prediction via natural gradients (Duan et al., 2020). However, most of these methods lack formal theoretical guarantees, with many relying primarily on heuristic justifications.

A principled route to uncertainty quantification is through statistical inference, e.g. via prediction intervals. However, this remains limited. In the Bayesian setting, Ustimenko et al. (2023) show randomized boosting converges to a kernel ridge regression and approximate Gaussian process posterior. Yet, they re-run the entire boosting procedure to generate even a single posterior sample. Malinin et al. (2021) suggest a virtual ensemble method, but do not provide formal guarantees.

*This work was done prior to employment at Amazon.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

In the frequentist setting, Zhou and Hooker (2022) propose a regularization scheme yielding a central limit theorem for boosting via convergence to kernel ridge regression. However, they do not provide proper confidence or prediction intervals, and their method recovers at most half the true signal (rescaling is suggested, but this amplifies errors). Despite these limitations, their framework remains the only foundation for frequentist inference in boosting, which we extend here.

Other literature. Recent advances for random forests yield principled statistical methods for uncertainty quantification via the construction of valid confidence intervals (Wager et al., 2014; Wager and Athey, 2017; Athey et al., 2018; Mentch and Hooker, 2016b), and interpretability via hypothesis tests for variable importance (Mentch and Hooker, 2016a; Coleman et al., 2022). Much of this is achieved through viewing a random forest as an adaptive kernel method (Friedberg et al., 2021; Athey et al., 2018). This also holds for various incarnations of boosting (Zhou and Hooker, 2022; Ustimenko et al., 2023). Despite these findings, the literature on uncertainty quantification and statistical inference for boosting is far sparser – a gap we fill by exploiting this equivalence.

Motivated by this gap, we study frequentist inference for uncertainty quantification in boosting under supervised nonparametric regression: $y = \boldsymbol{f}(\mathbf{x}) + \epsilon$, with subgaussian noise with variance σ^2. Our goal is to develop principled, statistically sound procedures for uncertainty quantification, outlined below.

Methodological contributions. In detail, we provide the following methodological contributions:

1. *Improvements via dropout:* We incorporate random dropout (Rashmi and Gilad-Bachrach, 2015) into the regularization procedure of Zhou and Hooker (2022). This yields an improved procedure (Algorithm 1) that achieves provably better performance by up to a factor of 4 in the asymptotic relative efficiency (ARE), and increased signal recovery by up to a factor of 2. By tuning the dropout rate in Algorithm 1, we smoothly bridge the vanilla Boulevard method (Zhou and Hooker, 2022) and a modified Random Forest (Breiman, 2001) that draws observation subsamples without replacement and considers the full covariate space at each split.
2. *Parallel boosting:* We construct a novel leave-one-out procedure for parallel boosting (Algorithm 2) that, when coupled with the above regularization procedure, improves the ARE by at least a factor of 4, and enjoys increased signal recovery by at least a factor of 2.
3. *Prediction, confidence, and reproduction intervals:* We leverage the central limit theorems our procedures enjoy to construct a variety of procedures for statistical inference. First, we conduct predictive inference via constructing prediction intervals on (potentially unseen) labels $\mathbf{y}$. We then construct confidence intervals for the underlying ground truth function $\boldsymbol{f}$. Lastly, to quantify uncertainty within the learning algorithm, we provide reproduction intervals (Zhou and Hooker, 2022) for another booster $\widehat{\boldsymbol{f}}$ trained on an independent realization of the training set.
4. *Hypothesis testing:* We construct a chi-squared hypothesis test for variable importance, extending the work of Mentch and Hooker (2016b) for random forests to the boosting setting.
5. *Computational efficiency:* Unlike prior approaches to statistical inference for random forests and boosting, we utilize the Nyström approximation of Musco and Musco (2017). This allows our tests and intervals to scale linearly with the number of data points.

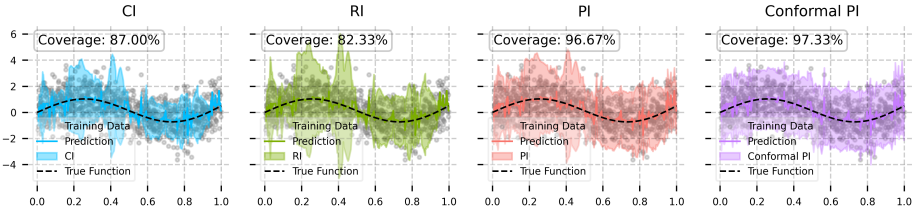

Figure 1: Demonstration of confidence, prediction, and reproduction intervals on $\boldsymbol{f}(x) = \sin 2\pi x + \frac{1}{2}x^2$. Conformal baseline. 200 trees, learning rate 0.6, depth 8, subsampling and dropout 0.6.

Theoretical contributions. Our primary theoretical contributions are central limit theorems (CLTs) for boosting with dropout (Algorithm 1) and parallel training (Algorithm 2) – the first such results for these settings to our knowledge. These extend the theoretical arguments of Zhou and Hooker (2022). In analyzing boosting with dropout, the stochasticity introduced into ensemble construction must be accounted for – in addition to randomness from data subsampling. To address this, we establish a weak law for finite-sample convergence in the stochastic contraction framework of Almudevar (2022). The arguments for parallel boosting are more challenging. Inspired by classical backfitting methods (Breiman and Friedman, 1985), we develop a delayed averaging scheme that

enables parallelism while preserving convergence guarantees. To achieve central limit theorems that hold unconditionally on the data $\mathbf{X}$, we show a bound on the maximal leaf size is sufficient to limit the influence of distant points and promote balanced splits, enabling rigorous distributional results.

Numerical experiments. We establish the efficacy of our methods on both a simulation study and real-world datasets. Our numerical experiments showcase that our algorithms can be tuned to interpolate between regularized boosting and random forests, and are highly competitive in terms of MSE. These also demonstrate the correctness, coverage, and computational efficiency of the statistical procedures we construct via the central limit theorems our algorithms enjoy.

2 Setup

Consider boosting for nonparametric regression with squared error loss. Given random features $\mathbf{x} \in [0,1]^d$, labels $y \in \mathbb{R}$, and noise $\epsilon \sim \text{SubG}(0, \sigma^2)$, say there exists some function $\boldsymbol{f}$ such that $y = \boldsymbol{f}(\mathbf{x}) + \epsilon$. The learner is given a training set $(\mathbf{X}^{\text{train}}, \mathbf{y}^{\text{train}})$ of size n, and learns an estimate $\widehat{\boldsymbol{f}}$ of $\boldsymbol{f}$ by minimizing the mean squared error (MSE) $\frac{1}{n}\sum_{i=1}^{n}(\widehat{\boldsymbol{f}}(\mathbf{x}_i) - y_i)^2$ over the training set. It is customary to analyze how $\widehat{\boldsymbol{f}}$ converges to $\boldsymbol{f}$, or in other words, the generalization error of $\widehat{\boldsymbol{f}}$ on a test set $(\mathbf{X}^{\text{test}}, \mathbf{y}^{\text{test}})$. Subscripts $\widehat{\boldsymbol{f}}_n^{(b)}$ indicate the estimated predictor trained on n datapoints for b boosting rounds, where we omit dependencies on b and n whenever possible.

Regularized stochastic gradient boosting. Zhou and Hooker (2022) propose a regularization procedure called Boulevard that recovers a central limit theorem for boosting. They do so by showing that the resulting regularized procedure converges to a kernel ridge regression. At each iteration

$$b = 1, ..., B-1, \text{ instead of } \widehat{\boldsymbol{f}}^{(b+1)} \leftarrow \widehat{\boldsymbol{f}}^{(b)} + \lambda \boldsymbol{t}^{(b)}, \text{ they update } \widehat{\boldsymbol{f}}^{(b+1)} \leftarrow \tfrac{b-1}{b}\widehat{\boldsymbol{f}}^{(b)} + \tfrac{\lambda}{b}\boldsymbol{t}^{(b)}$$

for the current function estimate $\widehat{\boldsymbol{f}}^{(b)}$, built tree $\boldsymbol{t}^{(b)}$, and learning rate $\lambda \in [0, 1)$. As this is equivalent to updating $\widehat{\boldsymbol{f}}^{(b+1)} \leftarrow \frac{\lambda}{b}\sum_{i=1}^{b}\boldsymbol{t}^{(i)}$, the resulting ensemble is an average of trees. This observation, along with assumptions on tree adaptivity, allows them to prove that Boulevard predictions are asymptotically normal, though they do not construct confidence or prediction intervals. Additionally, this regularization comes at a cost. It turns out that $\widehat{\boldsymbol{f}}_n^{(b)}$ converges only to $\frac{\lambda}{1+\lambda}\boldsymbol{f} \leq \boldsymbol{f}/2$ as $n, b \to \infty$, and so *Boulevard recovers at most half the signal*. Still, this is the only method available for frequentist inference for gradient boosting, and so we seek to improve upon it in this paper.

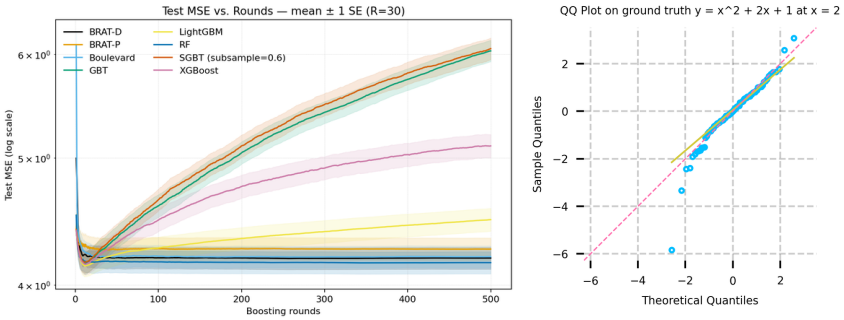

Figure 2: An added benefit of Boulevard regularization is that it renders early stopping unnecessary. Because the procedure converges to a fixed point, one can (in theory) boost indefinitely without encountering overfitting. It also produces asymptotically normal predictions (see the Q-Q plot).

Regression trees and tree structure. We are primarily concerned with boosting algorithms where we use regression trees as weak learners. We introduce a formulation of a regression tree below.

Definition 1 (Regression trees)**.** *A regression tree $\boldsymbol{t}_n$ trained on n datapoints segments the covariate space $[0,1]^d$ into a partition of hyper-rectangles $\{A_i\}_{i=1}^m$. When $A(\mathbf{x})$ for the rectangle containing $\mathbf{x}$, and $s_{n,j}(\mathbf{x})$ is the frequency at which training point $\mathbf{x}_j$ and test point $\mathbf{x}$ share a leaf:*

$$s_{n,j}(\mathbf{x}) = \frac{\mathbb{1}(\mathbf{x}_j \in A(\mathbf{x}))}{\sum_{k=1}^{n}\mathbb{1}(\mathbf{x}_k \in A(\mathbf{x}))}, \text{ the regression tree } \boldsymbol{t}_n \text{ predicts } \boldsymbol{t}_n(\mathbf{x}) = \sum_{j=1}^{n} s_{n,j}(\mathbf{x}) \cdot y_j.$$

As such, a tree can be thought of as an adaptive nearest neighbor method (Athey et al., 2018; Friedberg et al., 2021), where $s_{n,j}(\mathbf{x})$ yields the influence of datapoint $\mathbf{x}_j$ on the tree's prediction at test point $\mathbf{x}$. This motivates the definition of a structure vector and matrix from Zhou and Hooker (2022):

Definition 2 (Structure vectors and matrices)**.** *Let $\boldsymbol{t}_n$ be a tree trained on the input $(\mathbf{X}_n, \mathbf{y}_n)$. For an arbitrary point $\mathbf{x}$, let $\boldsymbol{s}_n(\mathbf{x}) = (s_{n,1}(\mathbf{x}), ..., s_{n,n}(\mathbf{x}))^\top$ be the structure vector of $\mathbf{x}$. As such, we say that the matrix $\mathbf{S}_n = (s_{n,j}(\mathbf{x}_i))_{i,j=1}^n$ with rows $\boldsymbol{s}_n(\mathbf{x}_i)^\top$ is the structure matrix of $\boldsymbol{t}_n$.*

The structure matrix is a kernel matrix. A natural idea is to perform kernel ridge regression (KRR):

$$\widehat{\boldsymbol{f}}_n(\mathbf{x}) = \langle \boldsymbol{r}_n(\mathbf{x}), \mathbf{y}_n \rangle,\ \boldsymbol{k}_n(\mathbf{x}) = \mathbb{E}[\boldsymbol{s}_n(\mathbf{x})],\ \mathbf{K}_n = \mathbb{E}[\mathbf{S}_n],\ \text{and } \boldsymbol{r}_n(\mathbf{x})^\top = \boldsymbol{k}_n(\mathbf{x})^\top(\lambda^{-1}\mathbf{I} + \mathbf{K}_n)^{-1}.$$

Zhou and Hooker (2022) show Boulevard regularization converges to a kernel ridge regression $\widehat{\boldsymbol{f}}_n(\mathbf{x}) = \langle \boldsymbol{r}_n(\mathbf{x}), \mathbf{y}_n \rangle$ as $b \to \infty$. However, as $\lambda \in (0, 1]$, this converges to at most half the signal. We later propose two algorithms in Algorithms 1 and 2, that converge to $\langle \boldsymbol{r}_n^D(\mathbf{x}), \mathbf{y}_n \rangle = \boldsymbol{k}_n(\mathbf{x})^\top(\lambda^{-1}\mathbf{I} + q\mathbf{K}_n)^{-1}\mathbf{y}_n$ and $\langle \boldsymbol{r}_n^P(\mathbf{x}), \mathbf{y}_n \rangle = \boldsymbol{k}_n(\mathbf{x})^\top(\mathbf{I} + (K-1)\mathbf{K}_n)^{-1}K\mathbf{y}_n$ respectively, achieving increased signal recovery. As a final note, to accommodate training $\boldsymbol{t}_n(\mathbf{x}, \mathcal{G})$ on a subsample $\mathcal{G} \subseteq \{1, ..., n\}$, we write $s_{n,j}(\mathbf{x}; \mathcal{G}) = \mathbb{1}(\mathbf{x}_j \in A, j \in \mathcal{G}) / \sum_{k=1}^n \mathbb{1}(\mathbf{x}_k \in A, k \in \mathcal{G})$, and $\boldsymbol{t}_n(\mathbf{x}; \mathcal{G}) = \sum_{j=1}^n s_{n,j}(\mathbf{x}; \mathcal{G}) \cdot y_j$. We suppress the dependence on subsamples when possible.

Integrity & adaptivity. We inherit the following two assumptions from Boulevard regularization:

Assumption 1 (Structure-value isolation)**.** *$(\boldsymbol{t}_n^{(b)})_{b=1}^B$ has $\boldsymbol{s}_n^{(b)}(\mathbf{x}) \perp\!\!\!\perp \mathbf{y}_n$ for all $b = 1, ..., B$.*

Assumption 2 (Non-adaptivity)**.** *There exists a probability measure $\mathcal{Q}_n$ on the space of tree structures and some b' so the tree structure at iteration $b = b', ..., B$ is an independent draw from $\mathcal{Q}_n$.*

These are required for theoretical guarantees. In practice, the first can be implemented via a stronger form of honesty (Wager and Athey, 2017) that we call *integrity*, refitting leaf values on an independent dataset after boosting. One can also randomly sample splits, or get them from a random forest trained on an independent sample. The second only requires non-adaptivity after a finite number of iterations, which may occur naturally (Zhou and Hooker, 2022). To enforce it, one can boost normally for a fixed number of steps and then sample tree structures from earlier iterations. Still, numerical experiments indicate that our algorithms perform well even without these adjustments. Our assumptions are given in a colleted format in the supplement.

3 Algorithms

To address the limitations of Boulevard regularization while retaining its statistical guarantees, we propose two improvements that achieve enhanced signal recovery and improved sample complexity:

Alg. 1: By incorporating random dropout (Rashmi and Gilad-Bachrach, 2015) in the ensemble.
Alg. 2: By growing trees in parallel within each round with a leave-one-out procedure.

The intuition is as follows. Surprisingly, increasing the dropout probability allows one to recover more of the signal. This is because each tree in the current ensemble is effectively downweighted during residual construction, and the new tree is fit on more of the signal. On the other hand, as the number of trees grown in parallel within each round grows, the resulting ensemble increasingly resembles the procedure of boosting random forests in sequence – allowing each round to recover more signal than a single tree can. We present both algorithms and explain them in detail below.

Algorithm 1 Boulevard Regularized Additive regression Trees – Dropout (BRAT-D)

Input: Features $\mathbf{X}$, labels $\mathbf{y}$, dropout rate $p = 1 - q$, subsample rate ξ, learning rate λ, boosting rounds B. Set $\widehat{\boldsymbol{f}}^{(0)}, \boldsymbol{t}^{(0)} \leftarrow 0$.
for $b = 1, ..., B-1$ **do**
  Subsample $\mathcal{S}_b \subseteq \{0, ..., b-1\}$ w.p. q.
  Subsample data $\mathcal{G}_b \subseteq \{1, ..., n\}$ w.p. ξ.
  $z_i \leftarrow y_i - \frac{\lambda}{b} \sum_{s \in \mathcal{S}_b} \boldsymbol{t}^{(s)}(\mathbf{x}_i)$.
  Construct tree $\boldsymbol{t}^{(b)}$ on $(\mathbf{x}_i, z_i)_{i \in \mathcal{G}_b}$.
  $\widehat{\boldsymbol{f}}^{(b+1)} \leftarrow \frac{b-1}{b}\widehat{\boldsymbol{f}}^{(b)} + \frac{\lambda}{b}\boldsymbol{t}^{(b)} = \frac{\lambda}{b}\sum_{i=1}^b \boldsymbol{t}^{(i)}$.
end for
Return final predictor $\frac{1+\lambda q}{\lambda}\widehat{\boldsymbol{f}}^{(B)}$.

Algorithm 2 BRAT - Parallel (BRAT-P)

Input: Features $\mathbf{X}$, labels $\mathbf{y}$, subsample ξ, trees/round K, rounds B. $\widehat{\boldsymbol{f}}^{(0)}, \boldsymbol{t}^{(0,k)} \leftarrow 0$.
Fit $\widehat{\boldsymbol{f}}^{(1)}, (\boldsymbol{t}^{(1,k)})_{k=1}^K$ with K boosting steps.
for $b = 2, ..., B-1$ **do**
  for $k = 1, ..., K$ **in parallel do**
    Subsample $\mathcal{G}_{b,k} \subseteq \{1, ..., n\}$ w.p. ξ.
    $z_{i,k} \leftarrow y_i - \frac{1}{b-1}\sum_{s=1}^{b-1}\sum_{l \neq k} \boldsymbol{t}^{(s,l)}(\mathbf{x}_i)$
    Construct tree $\boldsymbol{t}^{(b,k)}$ on $(\mathbf{x}_i, z_{i,k})_{i \in \mathcal{G}_{b,k}}$.
  end for
  Define $\widehat{\boldsymbol{f}}^{(b+1)} \leftarrow \frac{1}{b}\sum_{s=1}^b \sum_{k=1}^K \boldsymbol{t}^{(s,k)}$.
end for
Return final predictor $\widehat{\boldsymbol{f}}^{(B)}$.

Random dropout. Unlike vanilla boosting, Algorithm 1 computes residuals with both data (Friedman, 2002) and ensemble subsampling (Rashmi and Gilad-Bachrach, 2015). At each round $b = 1, ..., B-1$, each tree is subsampled with a dropout rate of $p \in [0, 1)$ (equivalently, with probability $q = 1 - p$). We write $\mathcal{S}_b \subseteq \{0, ..., b-1\}$ for the collection of subsampled tree indices at each round b. Likewise, data indices $\mathcal{G}_b \subseteq \{1, ..., n\}$ are subsampled with probability $\xi \in (0, 1]$.

As a result, each tree $\boldsymbol{t}^{(b)}$ is trained on the dataset $(\mathbf{x}_i, z_i)_{i \in \mathcal{G}_b}$, where the residuals are given by $z_i = y_i - \frac{1}{b} \sum_{s \in \mathcal{S}} \boldsymbol{t}^{(s)}(\mathbf{x}_i)$. Within the residual computation, we intentionally divide the sum of the subsampled ensemble's predictions by b instead of $|\mathcal{S}|$. This is done so each new tree is fit on more of the signal, allowing for increased signal recovery when applying Boulevard regularization.

We later show in Section 5 that the predictions of Algorithm 1 converge to a kernel ridge regression that enjoys a central limit theorem. Informally, for any test point $\mathbf{x}$, we have that:

$$\widehat{\mathbf{y}}_n^{(b)} \xrightarrow[b \to \infty]{a.s.} \left(\lambda^{-1}\mathbf{I} + q\mathbb{E}\left[\mathbf{S}_n\right]\right)^{-1} \mathbb{E}\left[\mathbf{S}_n\right] \mathbf{y}_n, \quad \|\boldsymbol{r}_n^D(\mathbf{x})\|_2^{-1} \left(\widehat{\boldsymbol{f}}_n^D(\mathbf{x}) - \tfrac{\lambda}{1+\lambda q} \boldsymbol{f}(\mathbf{x})\right) \xrightarrow[n \to \infty]{d} \mathcal{N}(0, \sigma^2),$$

for a kernel matrix induced by the tree ensemble $\mathbf{K}_n = \mathbb{E}[\mathbf{S}_n]$, and limit $\widehat{\boldsymbol{f}}_n^D = \lim_{b \to \infty} \widehat{\boldsymbol{f}}_n^{(b)}$ where $\widehat{\boldsymbol{f}}_n^D(\mathbf{x}) = \langle \boldsymbol{r}_n^D(\mathbf{x}), \mathbf{y}_n \rangle = \boldsymbol{k}_n(\mathbf{x})^\top (\lambda^{-1}\mathbf{I} + q\mathbf{K}_n)^{-1} \mathbf{y}_n$. $\|\boldsymbol{r}_n\|_2$ is the norm of kernel weights. Algorithm 1 encompasses two important special cases. When $\lambda = 1$ and $p \to 1$, this yields a random forest. When $p = 0$, Algorithm 1 corresponds to the original Boulevard algorithm (Zhou and Hooker, 2022). Tuning the dropout parameter p allows the user to interpolate between the two, while improving over Zhou and Hooker (2022) by up to a factor of $\frac{1+\lambda}{1+\lambda q} \in (1, 2]$ in signal recovery.

Parallel boosting. Algorithm 2 integrates a novel leave-one-out procedure and Boulevard regularization into the parallel boosting framework (Long and Servedio, 2011; Karbasi and Larsen, 2023; da Cunha et al., 2024). The algorithm is warm-started by performing K vanilla boosting iterations in the first round. At each subsequent round $b = 2, ..., B-1$, we grow K trees $(\boldsymbol{t}^{(b,k)})_{k=1}^K$ in parallel on the K datasets $((\mathbf{x}_i, z_{i,k})_{i \in \mathcal{G}_{b,k}})_{k=1}^K$. These consist of independently subsampled data indices $\mathcal{G}_{b,k} \subseteq \{1, ..., n\}$, and residuals that are computed by leaving one "column" out (or alternatively, one tree out per round): $z_{i,k} = y_i - \frac{1}{b-1} \sum_{s=1}^{b-1} \sum_{l \neq k} \boldsymbol{t}^{(s,l)}(\mathbf{x}_i)$. The resulting predictor is given by the regularized boosting update: $\widehat{\boldsymbol{f}}^{(b+1)} \leftarrow \frac{b-1}{b} \widehat{\boldsymbol{f}}^{(b)} + \frac{1}{b} \sum_{k=1}^K \boldsymbol{t}^{(b,k)} = \frac{1}{b} \sum_{s=1}^b \sum_{k=1}^K \boldsymbol{t}^{(s,k)}$.

Likewise, we later show in Section 5 that the predictions of Algorithm 2 also converge to a kernel ridge regression that enjoys a central limit theorem. Informally, for any test point $\mathbf{x}$:

$$\widehat{\mathbf{y}}_n^{(b)} \xrightarrow[b \to \infty]{a.s.} \left(\mathbf{I} + (K-1)\mathbb{E}\left[\mathbf{S}_n\right]\right)^{-1} K\mathbb{E}\left[\mathbf{S}_n\right] \mathbf{y}_n, \quad \|\boldsymbol{r}_n^P(\mathbf{x})\|_2^{-1} \left(\widehat{\boldsymbol{f}}_n^P(\mathbf{x}) - \boldsymbol{f}(\mathbf{x})\right) \xrightarrow[n \to \infty]{d} \mathcal{N}(0, \sigma^2),$$

with limit $\widehat{\boldsymbol{f}}_n^P = \lim_{b \to \infty} \widehat{\boldsymbol{f}}_n^{(b)}$ given by $\widehat{\boldsymbol{f}}_n^P(\mathbf{x}) = \langle \boldsymbol{r}_n^P(\mathbf{x}), \mathbf{y}_n \rangle = \boldsymbol{k}_n(\mathbf{x})^\top (\mathbf{I} + (K-1)\mathbf{K}_n)^{-1} K \mathbf{y}_n$. Algorithm 2 can be viewed as boosting a modified parallel backfitting algorithm (Breiman and Friedman, 1985)[2] that is warm-started with K vanilla boosting iterations.[3] When $K = 1, B \to \infty$, Algorithm 2 reduces to a random forest $\mathbb{E}\left[\mathbf{S}_n\right] \mathbf{y}_n$. On the other hand, Algorithm 2 reduces to vanilla boosting when $B = 1, K \to \infty$. Note that the CLT only holds for $1 < K < \infty$.

The attentive reader may observe that we do not divide the final predictions by K. This is because the combination of leaving one column out when fitting a tree and Boulevard averaging is sufficient to allow our procedure to stabilize.[4] Still, dividing by K yields a viable algorithm (deferred to future work) that can be thought of as boosting random forests each of K trees over B rounds, using the natural strategy of averaging all trees but the left-out column to form predictions.

4 Statistical Inference and Uncertainty Quantification for Boosting

We now introduce our methods for conducting statistical inference for boosting; the theoretical basis for these procedures is provided below. Our methods rest on the aforementioned convergence to a kernel ridge regression and central limit theorems that we formally prove later in Theorems 1 and 2. In the meantime, we provide the following asymptotically valid procedures.

[2]This can be thought of as a *structured* variant of dropout (Zhao et al., 2022; Xin et al., 2020; Fan et al., 2019). Dropping one out of every K trees allows the regularization to scale in $1/K$.

[3]The initial K boosting iterations yield better convergence in practice, and do not hinder our theoretical arguments as the impact of the vanilla boosting iterations washes out when we take $B \to \infty$.

[4]Parallelized backfitting can be thought of as parallelized coordinate descent. Averaging over the coordinate updates is sufficient for convergence. Dividing by b, as in Boulevard averaging, is sufficient when $b \geq K$.

Estimation of asymptotic variance. The CLTs for Algorithms 1 and 2 imply that one can construct confidence intervals for $\boldsymbol{f}$ with the following normal approximations:

$$\lambda^{-1}(1+\lambda q)\widehat{\boldsymbol{f}}_n^D(\mathbf{x}) \overset{d}{\approx} \mathcal{N}\left(\boldsymbol{f}(\mathbf{x}), \lambda^{-2}(1+\lambda q)^2\|\boldsymbol{r}_n^D(\mathbf{x})\sigma^2\|_2^2\right), \quad \widehat{\boldsymbol{f}}_n^P(\mathbf{x}) \overset{d}{\approx} \mathcal{N}\left(\boldsymbol{f}(\mathbf{x}), \sigma^2\|\boldsymbol{r}_n^P(\mathbf{x})\|_2^2\right).$$

As such, it remains to estimate $\boldsymbol{r}_n^D$, $\boldsymbol{r}_n^P$, and σ^2. We first estimate $\boldsymbol{k}_n(\mathbf{x})$ by the sample averages

$$\widehat{\boldsymbol{k}}_n^D(\mathbf{x}) = \tfrac{1}{B}\textstyle\sum_{b=1}^B s_{n,i}^{(b)}(\mathbf{x}, \mathcal{G}_b), \quad \widehat{\boldsymbol{k}}_n^P(\mathbf{x}) = \tfrac{1}{BK}\textstyle\sum_{b=1}^B\sum_{k=1}^K s_{n,i}^{(b,k)}(\mathbf{x}, \mathcal{G}_{b,k}), \tag{1}$$

where $s_{n,i}^{(b)}(\mathbf{x}; \mathcal{G}_b)$ is the fraction of subsampled datapoints in the same leaf $A_j^{(b)}$ as $\mathbf{x}$ over the trees. Likewise, we estimate $(\mathbf{K}_n)_{i,j}$, the fraction of trees where datapoints i and j fall in the same leaf:

$$(\widehat{\mathbf{K}}_n^D)_{i,j} = \tfrac{1}{B}\textstyle\sum_{b=1}^B s_{n,j}^{(b)}(\mathbf{x}_i; \mathcal{G}_b), \quad (\widehat{\mathbf{K}}_n^P)_{i,j} = \tfrac{1}{B}\textstyle\sum_{b=1}^B s_{n,j}^{(b,k)}(\mathbf{x}_i; \mathcal{G}_{b,k}), \tag{2}$$

Our estimates of $\widehat{\boldsymbol{r}}_n$ are then given by plugging the above into the formula for $\boldsymbol{r}_n$:

$$\widehat{\boldsymbol{r}}_n^D(\mathbf{x})^\top = \widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top(\lambda^{-1}\mathbf{I} + q\widehat{\mathbf{K}}_n^D)^{-1}, \quad \widehat{\boldsymbol{r}}_n^P(\mathbf{x})^\top = \widehat{\boldsymbol{k}}_n^P(\mathbf{x})^\top(\mathbf{I} + (K-1)\widehat{\mathbf{K}}_n^P)^{-1}K.$$

Lastly, we estimate σ^2 with the residuals $\widehat{\sigma}^2 = \frac{1}{n_{\text{cal}}}\sum_{i=1}^{n_{\text{cal}}}(y_i - \widehat{y}_i)^2$ on a hold-out calibration set. We describe a tweak later in Section 6 that uses the calibration set for increased robustness.

Confidence intervals for $\boldsymbol{f}$. Putting everything together yields $1-\alpha$ confidence intervals for $\boldsymbol{f}$ (where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the standard normal distribution):

$$\lambda^{-1}(1+\lambda q)\widehat{\boldsymbol{f}}_n^D(\mathbf{x}) \pm z_{1-\alpha/2}\lambda^{-1}(1+\lambda q)\widehat{\sigma}\|\widehat{\boldsymbol{r}}_n^D(\mathbf{x})\|_2, \quad \widehat{\boldsymbol{f}}_n^P(\mathbf{x}) \pm z_{1-\alpha/2}\widehat{\sigma}\|\widehat{\boldsymbol{r}}_n^P(\mathbf{x})\|_2. \tag{3}$$

Prediction intervals for $\mathbf{y}$. We can construct prediction intervals for $y|\mathbf{x}$ as follows:

$$\lambda^{-1}(1+\lambda q)\widehat{\boldsymbol{f}}_n^D(\mathbf{x}) \pm z_{1-\alpha/2}\lambda^{-1}(1+\lambda q)\widehat{\sigma}\sqrt{1+\|\widehat{\boldsymbol{r}}_n^D(\mathbf{x})\|_2^2}, \quad \widehat{\boldsymbol{f}}_n^P(\mathbf{x}) \pm z_{1-\alpha/2}\widehat{\sigma}\sqrt{1+\|\widehat{\boldsymbol{r}}_n^P(\mathbf{x})\|_2^2}.$$

These prediction intervals have asymptotic pointwise coverage, conditional on the test point $\mathbf{x}$. That is, $\mathbb{P}(y \in \text{PI}(\mathbf{x})|\mathbf{x}) \to 1-\alpha$ as $n \to \infty$, which holds as a corollary of the CLTs we present in the following section. Note that this conditional coverage guarantee is a stronger guarantee than is typically possible with conformal inference (Barber et al., 2020). We compare our prediction intervals to conformal prediction intervals within our numerical experiments later in this paper.

Reproduction intervals for $\widehat{\boldsymbol{f}}$. We now turn our attention towards computing reproduction intervals. That is, a confidence interval for another realization of $\widehat{\boldsymbol{f}}$ trained on another independent realization of the data. To do so, as Zhou and Hooker (2022) suggest, it suffices to scale the width of the above confidence intervals for $\boldsymbol{f}$ by $\sqrt{2}$ as $X_1, X_2 \overset{i.i.d.}{\sim} \mathcal{N}(0,\sigma^2) \implies X_1 - X_2 \sim \mathcal{N}(0, 2\sigma^2)$. Note that one can invert the above confidence, prediction, or reproduction intervals in order to test if the underlying function, another realization of the response, or average learner is equal to a given constant. This amounts to checking if the constant is contained in the interval of choice.

Variable importance tests. Suppose we wish to test for variable importance. Consider the possibility that there exists some subset of the feature space $[0,1]^d$ containing only $\breve{d} < d$ features, and let $\boldsymbol{g} : [0,1]^{\breve{d}} \to \mathbb{R}$ be the projection of $\boldsymbol{f}$ onto $\mathcal{L}^2([0,1]^{\breve{d}})$. We test the possibility that $\boldsymbol{f}(\mathbf{x}) = \boldsymbol{g}(\mathbf{x})$ by proxy. Consider a split of the training dataset $(\mathbf{X}_n, \mathbf{y}_n)$ into $(\mathbf{X}_{n,1}, \mathbf{y}_{n,1}), (\mathbf{X}_{n,2}, \mathbf{y}_{n,2})$, where $\mathbf{X}_{n,1} \in \mathbb{R}^{n/2,d}, \mathbf{X}_{n,2} \in \mathbb{R}^{n/2,\breve{d}}$. Let $\widehat{\boldsymbol{f}}_{n,1}$ be the boosting learner trained on $(\mathbf{X}_n, \mathbf{y}_n)$, and $\widehat{\boldsymbol{f}}_{n,2}$ be the same trained on $(\mathbf{X}_{n,2}, \mathbf{y}_{n,2})$. Given a hold-out dataset $(\mathbf{X}_m, \mathbf{y}_m)$, we test the null:

$$H_0 : \boldsymbol{f}(\mathbf{x}_j) = \boldsymbol{g}(\mathbf{x}_j) \text{ for all } j = 1, ..., m \text{ against } H_1 : \boldsymbol{f}(\mathbf{x}_j) \neq \boldsymbol{g}(\mathbf{x}_j) \text{ for some } j$$

by comparing $\widehat{\boldsymbol{f}}_{n,1}(\mathbf{x}_j)$ and $\widehat{\boldsymbol{f}}_{n,2}(\mathbf{x}_j)$. We exploit the CLTs our algorithms enjoy to do so. Write $\widehat{\boldsymbol{r}}_n^{D,P}(\mathbf{X}_m)^\top \in \mathbb{R}^{m\times n}$ for the matrix with rows $(\widehat{\boldsymbol{r}}_n^{D,P}(\mathbf{x}_l)^\top)_{l=1}^m$, where we write D, P within the superscript if the formula holds for both Algorithms 1 and 2. The difference in predictions

$$\widehat{\boldsymbol{d}}_m = (\widehat{\boldsymbol{r}}_{n,1}^{D,P}(\mathbf{X}_m) - \widehat{\boldsymbol{r}}_{n,2}^{D,P}(\mathbf{X}_m))^\top\mathbf{y} \overset{d}{\to} \mathcal{N}_m(\mathbf{0}, \sigma^2\widehat{\boldsymbol{\Xi}}_n) \text{ is multivariate normal under the null,}$$

with covariance matrix $\sigma^2\widehat{\boldsymbol{\Xi}}_n = \sigma^2(\widehat{\boldsymbol{r}}_{n,1}^{D,P}(\mathbf{X}_m) - \widehat{\boldsymbol{r}}_{n,2}^{D,P}(\mathbf{X}_m))^\top(\widehat{\boldsymbol{r}}_{n,1}^{D,P}(\mathbf{X}_m) - \widehat{\boldsymbol{r}}_{n,2}^{D,P}(\mathbf{X}_m))$ estimated by plugging in $\widehat{\sigma}^2$. As such, we can conduct a chi-squared test with test statistic

$$\widehat{\sigma}^{-2}\widehat{\boldsymbol{d}}_m^\top\widehat{\boldsymbol{\Xi}}_n^{-1}\widehat{\boldsymbol{d}}_m \sim \chi_m^2 \text{ under the null, rejecting the null if } \widehat{\sigma}^{-2}\widehat{\boldsymbol{d}}_m^\top\widehat{\boldsymbol{\Xi}}_n^{-1}\widehat{\boldsymbol{d}}_m > \chi_{m,1-\alpha}^2.$$

The test as presented runs in $O(n^3)$ time. In Appendix A, we present an accelerated version of the test that runs in $O(ns(r+s)+r^3)$ time, where s is the number of points subsampled for Nyström approximation and r is the number of test points subsampled.

Matrix sketching. Although the CLT reduces the problem of inference to kernel ridge regression (with the right kernel), computing $\widehat{\mathbf{K}}_n$ is an $O(n^3)$ problem – intractable for large n. This is the chief difficulty that Zhou and Hooker (2022) face when constructing replication intervals for vanilla Boulevard. We bypass this issue, making our procedures practical and tractable via matrix sketching.

We approximate $\widehat{\mathbf{K}}_n$ using the Nyström method (Williams and Seeger, 2000), via either uniform subsampling or the recursive approach of Musco and Musco (2017), producing an approximation $\widetilde{\mathbf{K}}_n \approx \widehat{\mathbf{K}}_n\mathbf{S}(\mathbf{S}^\top\widehat{\mathbf{K}}_n\mathbf{S})^\dagger\mathbf{S}^\top\widehat{\mathbf{K}}_n$, where $\mathbf{S} \in \mathbb{R}^{n\times s}$ is a random subsampling matrix. With the Nystrom method of Musco and Musco (2017), we can choose s to be near-linear in the effective dimension d_{eff}^{μ} of a ridge regression problem with regularization parameter μ on the kernel matrix: $s = \widetilde{O}(d_{\text{eff}}^{\mu})$. This requires only $O(ns^2)$ time to precompute the kernel, and only $O(s^2)$ time for inference – yielding practical statistical inference in near-linear time in the number of datapoints n. We adopt this approach in our experiments, allowing our procedures to run in linear time and remain practical, in contrast to previous work (Zhou and Hooker, 2022; Mentch and Hooker, 2016b). See Appendix A.

5 Theoretical Guarantees

We now formally present guarantees for convergence and asymptotic normality for Algorithms 1 and 2. Consider the (Lipschitz) nonparametric regression model introduced below:

Assumption 3 (Lipschitz Nonparametric Regression)**.** *Let $y = f(\mathbf{x}) + \epsilon$, where $\mathbf{x}$ has density μ satisfying $0 < c_1 \le \mu(\mathbf{x}) \le c_2 < \infty$ on its support. Further assume f is α-Lipschitz continuous on $\operatorname{supp}(\mu)$, and noise ϵ is sub-Gaussian with variance proxy σ^2.*

Finite-sample convergence to KRR. We now show that our algorithms achieve finite-sample convergence to kernel ridge regression as the number of boosting rounds increases. At first glance, this is surprising – greedy split selection and sequential ensemble construction create a highly nonlinear and time-dependent mapping. Fortunately, the regularization procedure of Zhou and Hooker (2022) provides a way forward. For the special case of $p = 0$ within Algorithm 1, they show convergence to a kernel ridge regression via the stochastic contraction framework of Almudevar (2022), utilizing Assumptions 1 (structure–value isolation) and 2 (non-adaptivity) to ensure stagewise stability. A key novelty within our analysis lies in showing that this convergence extends to Algorithms 1 and 2, even with the added complexity introduced by dropout and parallelism respectively. Despite these added sources of stochasticity and dependence, our use of the same regularization mechanism ensures that both algorithms inherit the stability required for convergence:

Theorem 1 (Finite Sample Convergence to KRR)**.** *For fixed $\mathbf{X}, \mathbf{y}$, under Assumptions 1, 2, and 3,*

$$\widehat{\mathbf{y}}_b \xrightarrow{a.s.} \left(\lambda^{-1}\mathbf{I} + q\mathbb{E}\left[\mathbf{S}_n\right]\right)^{-1}\mathbb{E}\left[\mathbf{S}_n\right]\mathbf{y} \text{ for Alg. 1}, \quad \widehat{\mathbf{y}}_b \xrightarrow{a.s.} \left(I + (K-1)\mathbb{E}\left[\mathbf{S}_n\right]\right)^{-1} K\mathbb{E}\left[\mathbf{S}_n\right]\mathbf{y} \text{ for Alg. 2},$$

as $b \to \infty$, where $q = 1 - p$ is one minus the dropout probability, K is the number of trees grown in parallel, and $\mathbf{S}_n$ is the kernel matrix induced by the tree structures within the ensemble.

The proof, deferred to Appendix C, departs from Zhou and Hooker (2022) in two crucial ways. Within Algorithm 1, dropout injects unbounded variance into the ensemble updates. To control this, we introduce the hard truncation function $\Gamma_M(y) = \operatorname{sign}(y)\min\{M, |y|\}$ into each residual update, $\widehat{\mathbf{y}}_b = \frac{b-1}{b}\widehat{\mathbf{y}}_{b-1} + \frac{\lambda}{b}\mathbf{S}_b\big(\mathbf{y} - \Gamma_M(\widetilde{\mathbf{y}}_b)\big)$, and then show that the probability of the partial ensemble escaping this cap vanishes as $b \to \infty$. Within Algorithm 2, truncating $\widetilde{\mathbf{y}}_{b,k} = \widehat{\mathbf{y}}_{b-1,K} - \Gamma_M(\frac{1}{b-1}\sum_{g=1}^{b-1}\widehat{\boldsymbol{t}}_{b,k})$ alone is insufficient. We further introduce a delay mechanism, requiring that $\boldsymbol{t}_{b,k}$ does not rely on $\boldsymbol{t}_{b,k-1}$, allowing us to apply Theorem 3 and parallelize tree training.

A central limit theorem. Having established almost-sure convergence to fixed points, we now examine the asymptotic distribution of our estimators to justify the statistical procedures introduced in Section 4. We seek a central limit theorem for the learner $\widehat{\boldsymbol{f}}_n^{D,P}$, trained on random $\mathbf{X}_n, \mathbf{y}_n$, demonstrating that the predictions $\widehat{\boldsymbol{f}}_n^{D,P}(\mathbf{x})$ are asymptotically normal with mean $\boldsymbol{f}(\mathbf{x})$ as $n \to \infty$. To do so, we inherit the following assumptions from Zhou and Hooker (2022): two on the leaves to control the norm of the KRR weight vector $\boldsymbol{r}_n^{D,P}$, and a restriction on the tree distribution space:

Assumption 4 (Bounded Leaf Diameter)**.** *Write $\mathsf{diam}(A) = \sup_{x,y\in A}\|x - y\|$. For any leaf A in a tree with structure $q \in Q_n$, we need $\sup_{A\in q}\mathsf{diam}(A) = O(d_n)$, where $d_n = O(n^{-1/(d+1)})$.*

Assumption 5 (Increased Minimal Leaf Size). *For any $\nu > 0$, $v_n = n^{-\frac{d+1}{d+2}+\nu} < n^{-\frac{d}{d+1}} = O(d_n^d)$.*

Assumption 6 (Restricted Tree Support). *The cardinality of the tree space Q_n is bounded by $O(n^{-1}\exp(0.5n^{1/(d+2)-\nu} - n^{\alpha}))$, for some small $\alpha > 0$.*

Intuitively, Assumption 5 prevents leaves from becoming too small too quickly, which in turn controls the maximal coordinate of the weight vectors across random samples. Assumption 6, guarantees that the complexity of possible tree partitions does not explode with n. We also make assumptions on the regularity of tree splits in line with the notion of α-regularity in Athey et al. (2018):

Assumption 7 (Median Splitting Rules). *The trees in Algorithm 2 are split at the medians.*

We defer the details to Definition 3. Under this splitting rule, the point-to-point collision probability is well-controlled, i.e. to say none of the leaves are big enough to contain most of the points. These conditions allow for uniform control over both leaf sizes and tree complexity, yielding the result:

Theorem 2 (Central Limit Theorem for Predictions). *Let $\mathbf{x} \in [0,1]^d$, $q \in (0,1]$, $K > 1$. As $n \to \infty$,*

$$\left\|\boldsymbol{r}_n^D\right\|_2^{-1}\left(\widehat{\boldsymbol{f}}_n^D(\mathbf{x}) - \lambda^{-1}(1+\lambda q)\boldsymbol{f}(\mathbf{x})\right) \xrightarrow{d} \mathcal{N}(0,\sigma^2), \quad \left\|\boldsymbol{r}_n^P\right\|_2^{-1}\left(\widehat{f}_n^P(\mathbf{x}) - f(\mathbf{x})\right) \xrightarrow{d} \mathcal{N}(0,\sigma^2).$$

We prove this in two stages. In the first, with proof deferred to Appendix D, we show asymptotic normality of the predictions around the noiseless KRR predictions $\langle \boldsymbol{r}_n^{D,P}(\mathbf{x}), \boldsymbol{f}(\mathbf{X}_n)\rangle$, conditional on the training data $\mathbf{X}_n, \mathbf{y}_n$ and test point $\mathbf{x}$, by applying the Lindeberg-Feller CLT to $\langle \mathbf{r}_n^{D,P}(\mathbf{x}), \varepsilon\rangle$. Next, we show that $\langle \boldsymbol{r}_n^{D,P}(\mathbf{x}), \boldsymbol{f}(\mathbf{X}_n)\rangle$ converges to the underlying ground truth function at a sufficiently fast rate in Appendices E and F. For Algorithm 1, although $\boldsymbol{r}_n^D$ now involves $[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n]^{-1}$, this still preserves the rate $\|\boldsymbol{r}_n\|_2 \asymp n^{-1/(2(d+1))}$, allowing us to show $\langle \mathbf{r}_n, f(\mathbf{X}_n)\rangle - \frac{\lambda q}{1+\lambda q}\boldsymbol{f}(\mathbf{x}) = o_p(\|\boldsymbol{r}_n\|_2)$. The Lindeberg–Feller verification then proceeds identically to the non-dropout case. However, to give a CLT for Algorithm 2, controlling the influence from distant training points is challenging, though we show that Assumption 7 is sufficient to do so.

Understanding the result. The rate of convergence of the CLT in Theorem 2 depends on the norm of $\boldsymbol{r}_n^{D,P}$. This is controlled as follows, yielding generalization bounds for Algorithms 1 and 2:

Lemma 1 (Rate of Convergence). *Let $\mathbf{B}_n := \{i : \|\mathbf{x} - \mathbf{x}_i\| \le d_n\}$ be the points within distance d_n from test point $\mathbf{x}$. If $|\mathbf{B}_n| = \Omega\left(n \cdot d_n^d\right)$, then $\|\mathbf{k}_n\|_2 = \Theta(n^{-\frac{1}{2}\frac{1}{d+1}})$, $\left\|\boldsymbol{r}_n^{D,P}\right\|_2 = \Theta(n^{-\frac{1}{2}\frac{1}{d+1}})$.*

Corollary 1. $\mathbb{E}[(\frac{1+\lambda q}{\lambda}\widehat{\boldsymbol{f}}_n^D(\mathbf{x}) - \boldsymbol{f}(\mathbf{x}))^2] \lesssim (\frac{1+\lambda q}{\lambda})^2\sigma^2 n^{-\frac{1}{d+1}}$ *and* $\mathbb{E}[(\widehat{f}_n^P(\mathbf{x}) - f(\mathbf{x}))^2] \lesssim \sigma^2 n^{-\frac{1}{d+1}}$.

Corollary 2. *If $n \ge \Omega\left(\frac{\log(1/\delta)}{\epsilon^{2d+2}}\right)$, $b \ge \Omega\left(\frac{n\lambda^2 M^2}{\epsilon^2\delta}\right)$, then w.p. at least $1-\delta$, $|\widehat{\boldsymbol{f}}_n^{(b)}(\mathbf{x}) - \boldsymbol{f}(\mathbf{x})| \le \epsilon$.*

The first is an asymptotic risk bound, while the second is a nonasymptotic PAC guarantee. Recall that M is the truncation level. The results above recover the minimax rate for nonparametric regression on $1/2$-Holder smooth functions (Stone, 1982) – quadratically worse than the rate for Lipschitz functions (the setting we are in). We believe this can be improved – trees should inherit adaptivity to the intrinsic dimension as an adaptive nearest neighbor method, but we leave this for future work.

Furthermore, Theorem 2 yields asymptotic coverage of our intervals constructed in Section 4:

Corollary 3. *The CIs, PIs and RIs achieve $1-\alpha$ almost sure pointwise coverage as $n, b \to \infty$.*

We show in Lemma 1 that the rate at which $\|\boldsymbol{r}_n^{D,P}\|_2$ grows is the same for Algorithms 1 and 2. This shows that our methods achieve an asymptotic relative efficiency relative to Zhou and Hooker (2022) by up to a factor of 4 for Algorithm 1, and at least a factor of 4 for Algorithm 2:

Corollary 4 (Asymptotic Relative Efficiency). *Write $\widehat{\boldsymbol{f}}^B$ for the scaled predictions made by vanilla Boulevard (Zhou and Hooker, 2022) with the same learning rate $\lambda \in (0,1]$ as Algorithm 1. Then,*

$$\mathit{Var}(\tfrac{1+\lambda}{\lambda}\widehat{\boldsymbol{f}}^B)/\mathit{Var}(\tfrac{1+\lambda q}{\lambda}\widehat{\boldsymbol{f}}^D) = (\tfrac{1+\lambda}{1+\lambda q})^2 \in [1,4],\ \mathit{Var}(\tfrac{1+\lambda}{\lambda}\widehat{\boldsymbol{f}}^B)/\mathit{Var}(\widehat{\boldsymbol{f}}^P) = \left(\tfrac{1+\lambda}{\lambda}\right)^2 \in [4,\infty).$$

This has the implication that Algorithms 1 and 2 achieve increased signal recovery in the square root of the ARE relative to vanilla Boulevard. The intuition for this improvement is as follows. For any choice of λ within vanilla Boulevard, there exists a choice of p within Algorithm 1 that requires a smaller rescaling. On the other hand, as Algorithm 2 requires no rescaling, it can achieve an unbounded improvement in relative efficiency over vanilla Boulevard as we take $\lambda \to 0$.

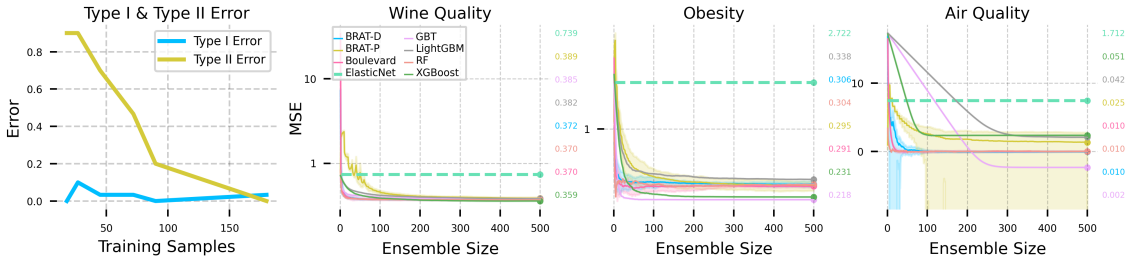

Figure 3: **Left:** Type I and Type II error of the hypothesis test for variable importance against training set size. Test set is of the same size. Error rates computed over 30 trials. **Right:** Comparison of MSE achieved by various machine learning algorithms on different datasets, with hyperparameters tuned for each by Optuna. Shaded area depicts two standard deviations over 5 trials.

6 Numerical Experiments

The previous section presents a rich theory; alongside convergence to kernel ridge regression and a central limit theorem for predictions, we provide asymptotic risk bounds and nonasymptotic PAC guarantees, and show asymptotic coverage for our intervals. In light of this, we present a series of numerical experiments to empirically justify our algorithms and methods.

Predictive accuracy. The first, and most natural, thing to do is to examine the performance of our algorithms against a handful of competitors in terms of test MSE on nine datasets from the UCI Machine Learning Repository in Figure 3 and Figure 6 in the supplement. Our competitors include popular methods such as XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), as well as random forests and vanilla boosting. We also compare to vanilla Boulevard (Zhou and Hooker, 2022), and an elastic net regression as a baseline. All hyperparameters were tuned with Optuna (Akiba et al., 2019), reported in Appendix I. There is no clear winner in general, but XGBoost is a consistently strong performer, and random forests are clearly more suited to some datasets than others. Neither Algorithm 1 nor Algorithm 2 consistently outperform each other, but Algorithm 2 occasionally exhibits instability on some datasets. Regardless, both of our algorithms are competitive in terms of final MSE and rate of convergence, and exhibit the ability to be tuned to be closer to boosting (Wine Quality) or random forests (Air Quality), providing increased flexibility by interpolating between the two.

Our results in Figure 3 are somewhat unfair to Algorithm 2, as the x-axis is in the ensemble size and not the number of boosting rounds. The number of boosting rounds that Algorithm 2 encounters can be as few as 31 on Abalone, in contrast to the 500 boosting rounds all other algorithms enjoy. Given the same number of boosting rounds, we would certainly expect an improvement.

Variable importance tests. We examine their performance by conducting a simulation study, testing the null $H_0 : w = 0$ on data from $\boldsymbol{f}(\mathbf{x}) = 4x_1 - x_2^2 + wbx_3$. We fit Algorithm 1 on data generated from $\boldsymbol{f}(\mathbf{x})$ and $\boldsymbol{g}(\mathbf{x}) = 4x_1 - x_2^2$, with 100 trees, $\lambda = 1$, subsampling rate 1, dropout rate 0.95, and a max depth of 6. The size and power of our test presented in Section 4 are depicted in the left panel within Figure 3. Our test performs very well, maintaining appropriate size control throughout while the Type II error decreases quickly. Empirically, increasing the dropout rate increases power.

Coverage of intervals. Other than the Nyström approximations for fast and practical interval computation in Appendix A, there is one more tweak that we encourage users to make in practice. We demonstrate in Corollary 3 that our intervals have asymptotic coverage, but this says nothing about its finite sample properties. Fortunately, Romano et al. (2020) yields a simple enhancement. Since we estimate $\widehat{\sigma}$ through computing the residuals on a hold-out calibration set, we reuse the calibration set to adaptively grow or shrink our intervals according to the empirical prediction interval coverage on the calibration set. This procedure converges quickly (amounting to a doubling trick and binary search), and empirically is very helpful in increasing robustness and finite-sample performance.

To examine the empirical performance of our intervals, we consider the Friedman function $\boldsymbol{f}(\mathbf{x}) = 10\sin(\pi \mathbf{x}_1 \mathbf{x}_2) + 20(\mathbf{x}_3 - 0.5)^2 + 5\mathbf{x}_5 - 10$ (Friedman, 2000). This is depicted as a raincloud plot in Figure 4. Importantly, we consider two notions of coverage: marginal coverage, where coverage is averaged over test points, and conditional coverage, where the coverage is conditional on a test point. The first two rows depict the former, while the last two depict the latter. Our intervals perform reasonably well, attaining nominal coverage with the exception of the reproduction interval's coverage and the confidence interval's conditional coverage. In particular, the prediction interval performs

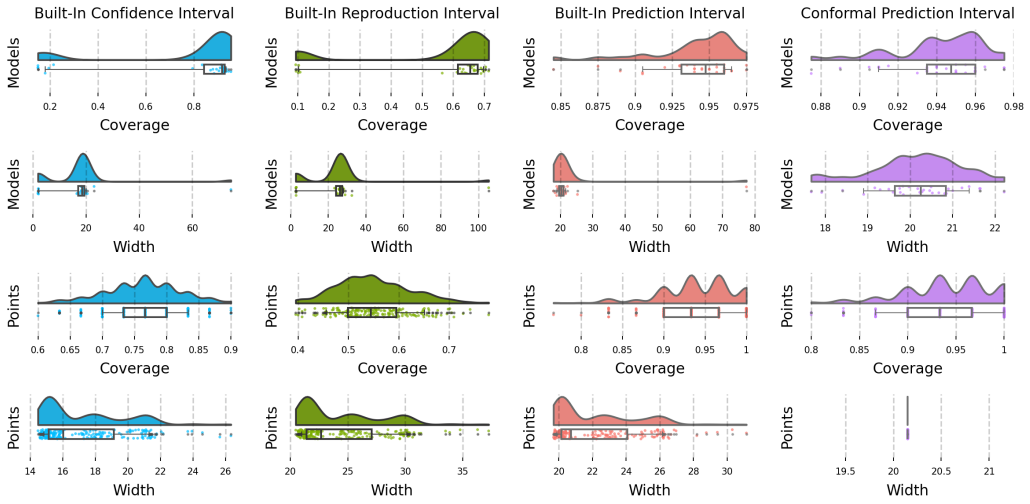

Figure 4: Points in the first two rows represent one of 30 models, depicting the fraction of test points that fell inside the interval generated by the model (marginal coverage) and average width for each model. Points in the last two represent one of 100 test points, depicting the fraction of models with intervals containing the test point (conditional coverage) and average width for each point. Results from Algorithm 1 with 200 trees, learning rate 0.6, subsampling 0.8, dropout 0.3, max depth 4.

very well after the adaptive coverage adjustment. Importantly, unlike the conformal benchmark that has a constant width at each point, our prediction intervals have different interval widths at each point, allowing users to identify 'hard" examples. See Appendix J for other hyperparameter choices.

CI tuning. Reducing subsample rate ξ inflates variance proportionally, while deeper trees yield narrower intervals as finer splits yield sparser $\boldsymbol{k}_n(\mathbf{x})$ and smaller $\|\boldsymbol{r}_n\|$. Variance grows with shrinkage $1/\lambda$ and falls as dropout p increases. Choosing moderate λ and increasing p yields stabler CIs. Setting $p \to 0$ empirically leads to wider intervals and less power in the hypothesis test for variable importance. This can be seen in Figure 5.

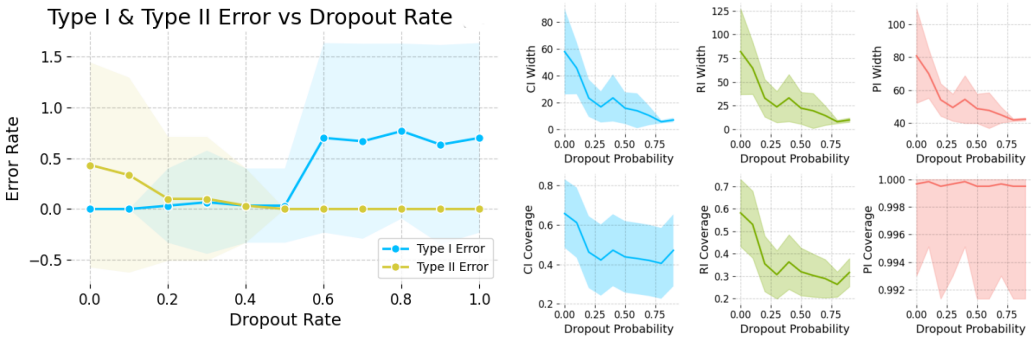

Figure 5: Comparison of size and power of statistical inference procedures against dropout probability. Conformal adjustment is not performed (for clearer benchmarking, hence the under/overcoverage), and we plot the coverage and width across all datapoints and repetitions.

7 Conclusions and Further Work

We incorporate dropout and parallel boosting into the regularization procedure of Zhou and Hooker (2022), allowing for increased signal recovery and improved asymptotic variance. With the CLTs our algorithms enjoy, we construct confidence and prediction intervals, and hypothesis tests for variable importance. Numerical experiments empirically validate our algorithms and statistical procedures.

Our theoretical guarantees depend on structure–value isolation, non-adaptivity, and tree regularity assumptions. Relaxing these – e.g., via honesty-based splitting or quantifying mild adaptivity – would broaden applicability. Extending inference beyond regression (to classification, survival, or structured outputs) also poses new challenges: non-quadratic losses, different tree behavior, and the need for new CLTs. Doing so under weaker assumptions is a welcome direction for future work.

References

(2019). Estimation of Obesity Levels Based On Eating Habits and Physical Condition . UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5H31Z.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Almudevar, A. (2022). A stochastic contraction mapping theorem.

Athey, S., Tibshirani, J., and Wager, S. (2018). Generalized random forests.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2020). The limits of distribution-free conditional predictive inference.

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

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598.

Brophy, J. and Lowd, D. (2022). Instance-based uncertainty estimation for gradient-boosted regression trees. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 11145–11159. Curran Associates, Inc.

Chen, M. and Wang, D. (2023). Uncertainty estimation for gradient boosting models based on Gaussian graph model and natural gradient. In Ba, S. and Zhou, F., editors, *Third International Conference on Machine Learning and Computer Application (ICMLCA 2022)*, volume 12636, page 1263633. International Society for Optics and Photonics, SPIE.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM.

Coleman, T., Peng, W., and Mentch, L. (2022). Scalable and efficient hypothesis testing with random forests. *J. Mach. Learn. Res.*, 23(1).

Cortez, P. (2008). Student Performance. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5TG7T.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Wine Quality. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56S3T.

da Cunha, A., Høgsgaard, M. M., and Larsen, K. G. (2024). Optimal parallelization of boosting.

Duan, T., Avati, A., Ding, D. Y., Thai, K. K., Basu, S., Ng, A. Y., and Schuler, A. (2020). Ngboost: Natural gradient boosting for probabilistic prediction.

Fan, A., Grave, E., and Joulin, A. (2019). Reducing transformer depth on demand with structured dropout.

Friedberg, R., Tibshirani, J., Athey, S., and and, S. W. (2021). Local linear forests. *Journal of Computational and Graphical Statistics*, 30(2):503–517.

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378.

Karbasi, A. and Larsen, K. G. (2023). The impossibility of parallelizing boosting.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Long, P. and Servedio, R. (2011). Algorithms and hardness results for parallel large margin learning. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Malinin, A., Prokhorenkova, L., and Ustimenko, A. (2021). Uncertainty in gradient boosting via ensembles.

Mentch, L. and Hooker, G. (2016a). Formal hypothesis tests for additive structure in random forests.

Mentch, L. and Hooker, G. (2016b). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17(26):1–41.

Musco, C. and Musco, C. (2017). Recursive sampling for the nystrom method. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Nash, W., Sellers, T., Talbot, S., Cawthorn, A., and Ford, W. (1994). Abalone. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55C7W.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6639–6649, Red Hook, NY, USA. Curran Associates Inc.

Rashmi, K. V. and Gilad-Bachrach, R. (2015). Dart: Dropouts meet multiple additive regression trees.

Redmond, M. (2002). Communities and Crime. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C53W3X.

Romano, Y., Sesia, M., and Candes, E. (2020). Classification with valid and adaptive coverage. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3581–3591. Curran Associates, Inc.

Schlimmer, J. (1985). Automobile. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5B01C.

Stone, C. J. (1982). Optimal Global Rates of Convergence for Nonparametric Regression. *The Annals of Statistics*, 10(4):1040 – 1053.

Tan, T., Huertas, C., and Zhao, Q. (2023). Efficient and effective uncertainty quantification in gradient boosting via cyclical gradient mcmc.

Ustimenko, A., Beliakov, A., and Prokhorenkova, L. (2023). Gradient boosting performs gaussian process inference.

Ustimenko, A. and Prokhorenkova, L. (2022). Sglb: Stochastic gradient langevin boosting.

Vito, S. (2008). Air Quality. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C59K5F.

Wager, S. and Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests.

Wager, S., Hastie, T., and Efron, B. (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*, 15(48):1625–1651.

Williams, C. and Seeger, M. (2000). Using the nyström method to speed up kernel machines. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Xin, J., Tang, R., Lee, J., Yu, Y., and Lin, J. (2020). Deebert: Dynamic early exiting for accelerating bert inference.

Yin, X., Fallah-Shorshani, M., McConnell, R., Fruin, S., Chiang, Y.-Y., and Franklin, M. (2023). Quantile extreme gradient boosting for uncertainty quantification.

Zhao, Y., Dada, O., Gao, X., and Mullins, R. D. (2022). Revisiting structured dropout.

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

A Matrix Sketching

We first sketch the kernel inversion where we approximate $\widehat{\mathbf{K}}_n^D \approx \widetilde{\mathbf{K}}_n^D = \widehat{\mathbf{K}}_n^D\mathbf{S}(\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^\dagger\mathbf{S}^\top\widehat{\mathbf{K}}_n^D$:

$$\begin{aligned}
\widehat{\boldsymbol{r}}_n^D(\mathbf{x})^\top\mathbf{y}_n &= \widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top(\lambda^{-1}\mathbf{I} + q\widehat{\mathbf{K}}_n^D)^{-1}\mathbf{y}_n \\
&\approx \widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top(\lambda^{-1}\mathbf{I} + q\widetilde{\mathbf{K}}_n^D)^{-1}\mathbf{y}_n \\
&= \widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top(\lambda^{-1}\mathbf{I} + q\widehat{\mathbf{K}}_n^D\mathbf{S}(\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^\dagger\mathbf{S}^\top\widehat{\mathbf{K}}_n^D)^{-1}\mathbf{y}_n \\
&= \widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top\left(\lambda\mathbf{I} - \lambda^2\widehat{\mathbf{K}}_n^D\mathbf{S}(q^{-1}\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S} + \lambda(\widehat{\mathbf{K}}_n^D\mathbf{S})^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^{-1}\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\right)\mathbf{y}_n \\
&= \widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top\widehat{\boldsymbol{\Lambda}}_n^D\mathbf{y}_n = \widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top\widehat{\boldsymbol{\alpha}}_n,
\end{aligned}$$

where we define the n-dimensional coefficient vector

$$\widehat{\boldsymbol{\alpha}}_n = \left(\lambda\mathbf{I} - \lambda^2\widehat{\mathbf{K}}_n^D\mathbf{S}(q^{-1}\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S} + \lambda(\widehat{\mathbf{K}}_n^D\mathbf{S})^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^{-1}\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\right)\mathbf{y}_n,$$

and write

$$\widehat{\boldsymbol{\Lambda}}_n^D = \left(\lambda\mathbf{I} - \lambda^2\widehat{\mathbf{K}}_n^D\mathbf{S}(q^{-1}\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S} + \lambda(\widehat{\mathbf{K}}_n^D\mathbf{S})^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^{-1}\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\right) \in \mathbb{R}^{n\times n}$$

for our sketched estimate of the inverse kernel matrix. Computing this takes $O(ns^2)$ time when $\mathbf{S}$ is a subsampling matrix (Musco and Musco, 2017).

Now we further sketch the coefficients $\widetilde{\boldsymbol{\alpha}}_n = (\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^\dagger\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\widehat{\boldsymbol{\alpha}}_n \in \mathbb{R}^s$, in line with Appendix C of Musco and Musco (2017):

$$\widehat{\boldsymbol{k}}_n^D(\mathbf{x})^\top\widehat{\boldsymbol{\alpha}}_n \approx \left\langle \mathbf{S}^\top\widehat{\boldsymbol{k}}_n^D(\mathbf{x}), (\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^\dagger\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\widehat{\boldsymbol{\alpha}}_n \right\rangle = \left\langle \widetilde{\boldsymbol{k}}_n^D(\mathbf{x}), \widetilde{\boldsymbol{\alpha}}_n \right\rangle.$$

When $\widetilde{\boldsymbol{\alpha}}_n = (\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^\dagger\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\widehat{\boldsymbol{\alpha}}_n$ is precomputed, we can make a new KRR point prediction in only $O(s)$ time. This is because we only need to compute the s coordinates of the vector $\widetilde{\boldsymbol{k}}_n^D(\mathbf{x}) = \mathbf{S}^\top\widehat{\boldsymbol{k}}_n^D(\mathbf{x}) \in \mathbb{R}^s$, taking s kernel evaluations, and multiplication with $\widetilde{\boldsymbol{\alpha}}_n = (\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^\dagger\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\widehat{\boldsymbol{\alpha}}_n \in \mathbb{R}^s$ can be done in $O(s)$ time.

$O(s^2)$ time inference. So in order to perform inference in sublinear time, we first precompute in $O(ns^2)$ time:

$$\widetilde{\boldsymbol{\Lambda}}_n^D = \widehat{\boldsymbol{\Lambda}}_n^D\widehat{\mathbf{K}}_n^D\mathbf{S}(\mathbf{S}^\top\widehat{\mathbf{K}}_n^D\mathbf{S})^\dagger \in \mathbb{R}^{n\times s}, \qquad \widehat{\boldsymbol{\Sigma}}_n^D = \left(\widetilde{\boldsymbol{\Lambda}}_n^D\right)^\top\widetilde{\boldsymbol{\Lambda}}_n^D \in \mathbb{R}^{s\times s}.$$

To obtain a sketched estimate of $\|\boldsymbol{r}_n^D(\mathbf{x})\|_2$ on a new test point $\mathbf{x}$, we compute the s coordinates of the vector $\widetilde{\boldsymbol{k}}_n^D(\mathbf{x}) = \mathbf{S}^\top\widehat{\boldsymbol{k}}_n^D(\mathbf{x}) \in \mathbb{R}^s$, and obtain an estimate

$$\|\widetilde{\boldsymbol{r}}_n^D(\mathbf{x})\|_2 = \sqrt{\widetilde{\boldsymbol{k}}_n^D(\mathbf{x})^\top\widehat{\boldsymbol{\Sigma}}_n^D\widetilde{\boldsymbol{k}}_n^D(\mathbf{x})}$$

in $O(s^2)$ time. The equivalent expression $\|\widetilde{\boldsymbol{r}}_n^D(\mathbf{x})\|_2 = \|\widetilde{\boldsymbol{\Lambda}}_n^D\widetilde{\boldsymbol{k}}_n^D(\mathbf{x})\|_2$ takes $O(n)$ time.

Sketched hypothesis testing. Given a hold-out dataset $(\mathbf{X}_m, \mathbf{y}_m)$, we test the null hypothesis:

$$H_0 : \boldsymbol{f}(\mathbf{x}_j) = \boldsymbol{g}(\mathbf{x}_j) \text{ for all } j = 1, ..., m \text{ against } H_1 : \boldsymbol{f}(\mathbf{x}_j) \neq \boldsymbol{g}_n(\mathbf{x}_j) \text{ for some } j$$

by comparing $\widehat{\boldsymbol{f}}_{n,1}(\mathbf{x}_j), \widehat{\boldsymbol{f}}_{n,2}(\mathbf{x}_j)$. However, given m test points on a test dataset $(\mathbf{X}_m, \mathbf{y}_m)$, we can subsample r points with a new subsampling matrix $\boldsymbol{r}$ to compute the doubly subsampled kernel

No.	Condition	Use	How Satisfied
1	Tree structure $\boldsymbol{s}_n^{(b)}(\mathbf{x}) \perp\!\!\!\perp \mathbf{y}_n$ for all $b = 1, ..., B$	Theorem 1; allows kernel ridge regression representation	Random tree structures, sample splitting
2	Trees eventually i.i.d. $\boldsymbol{s}_n^b \sim \mathrm{Q}_n$	Theorem 1; ensures averaging tree structures has a limit	Random tree structures, draw from library of trees.
3	f α-Lipschitz, $\mu(x)$ bounded away from 0	Theorem 2; controls ability to approximate f by kernel ridge regression	Structural assumption about data generating process
4	Shrinking maximum leaf diameter $\sup_{A \in Q_n} \mathrm{diam}(A)$ bounded by $O(n^{-1/(d+1)})$	Theorem 2; controls within-leaf bias	Minimum probability of splitting each feature
5	Growing observations per leaf $v_n = n^{\frac{d+1}{d+1}+\nu} < n^{\frac{d}{d+1}} = O(d_n^d)$	Theorem 2; controls variance within leaves	Minimum leaf size
6	Cardinality of tree space $\lvert Q_n \rvert$ bounded by $O(n^{-1} \exp(0.5n^{1/(d+2)-\nu} - n^{\alpha}))$	Theorem 2; restricts variance of models	Restrict tree depth, choose library of trees
7	Trees Split on Medians	Theorem 2; Alg 2 only; controls bias and variance	Restrict candidate splits

Table 1: Table of assumptions required for our theory and their use.

matrix $\widetilde{\boldsymbol{\kappa}}_{n,1}^D(\mathbf{X}_m) = \mathbf{S}^\top \widehat{\boldsymbol{k}}_n^D(\mathbf{X}_m)\boldsymbol{r} \in \mathbb{R}^{s \times r}$, which we only need to perform $s \times r$ kernel operations to compute. So the difference statistic is then given by

$$\widetilde{\boldsymbol{d}}_m = \widetilde{\boldsymbol{\kappa}}_{n,1}(\mathbf{X}_m)^\top \widetilde{\boldsymbol{\alpha}}_{n,1} - \widetilde{\boldsymbol{\kappa}}_{n,2}(\mathbf{X}_m)^\top \widetilde{\boldsymbol{\alpha}}_{n,2} = \left(\widetilde{\boldsymbol{\Lambda}}_{n,1}^D \widetilde{\boldsymbol{\kappa}}_{n,1}(\mathbf{X}_m) - \widetilde{\boldsymbol{\Lambda}}_{n,2}^D \widetilde{\boldsymbol{\kappa}}_{n,2}(\mathbf{X}_m)\right)^\top \mathbf{y} \in \mathbb{R}^r.$$

Modulo subsampling and Nyström approximation error, this is multivariate normal under the null: $\widetilde{\boldsymbol{d}}_m^D \sim \mathcal{N}_r\left(\mathbf{0}, \sigma^2 \widetilde{\boldsymbol{\Xi}}_n^D\right)$, with covariance matrix

$$\widetilde{\boldsymbol{\Xi}}_n^D = \left(\widetilde{\boldsymbol{\Lambda}}_{n,1}^D \widetilde{\boldsymbol{\kappa}}_{n,1}(\mathbf{X}_m)^\top - \widetilde{\boldsymbol{\Lambda}}_{n,2}^D \widetilde{\boldsymbol{\kappa}}_{n,2}(\mathbf{X}_m)^\top\right)^\top \left(\widetilde{\boldsymbol{\Lambda}}_{n,1}^D \widetilde{\boldsymbol{\kappa}}_{n,1}(\mathbf{X}_m)^\top - \widetilde{\boldsymbol{\Lambda}}_{n,2}^D \widetilde{\boldsymbol{\kappa}}_{n,2}(\mathbf{X}_m)^\top\right) \in \mathbb{R}^{r \times r}.$$

We then have the test statistic:

$$\widehat{\sigma}^{-2} \widetilde{\boldsymbol{d}}_m^{D\top} (\widetilde{\boldsymbol{\Xi}}_n^D)^{-1} \widetilde{\boldsymbol{d}}_m^D \sim \chi_r^2.$$

The runtime of the test is as follows. It takes $O(ns^2)$ precomputation time to form $\widetilde{\boldsymbol{\Lambda}}_{n,1}^D, \widetilde{\boldsymbol{\Lambda}}_{n,2}^D \in \mathbb{R}^{n \times s}$, $O(nsr)$ time for the multiplication with $\widetilde{\boldsymbol{\kappa}}_{n,1}(\mathbf{X}_m), \widetilde{\boldsymbol{\kappa}}_{n,2}(\mathbf{X}_m)$, and $O(n)$ time for multiplication with $\mathbf{y}$. Once one has the noise estimate, inverting the covariance matrix and computing the test points takes $O(r^3)$ time.

B Table of Assumptions

Table 1 provides a summary of the assumptions we require, a brief explanation of how they are used and conditions under which they are satisfied.

C Finite Sample Convergence

C.1 Stochastic Contraction Mapping Theorem

To show convergence for both algorithms, we will show both of them are a stochastically contracted process. The criteria of deciding such process can be found in Almudevar (2022), and we introduce the theorem below.

Theorem 3. *Given $\mathbb{R}^d$-valued stochastic process $\{\mathbf{z}_t\}_{t\in\mathbb{N}}$, a sequence of $0 < \lambda_t \leq 1$, define*

$$\mathcal{F}_0 = \emptyset, \mathcal{F}_t = \sigma(\mathbf{z}_1, \ldots, \mathbf{z}_t),$$
$$\epsilon_t = \mathbf{z}_t - \mathbb{E}[\mathbf{z}_t|\mathcal{F}_{t-1}].$$

We call $\mathbf{z}_t$ a stochastic contraction if the following properties hold

1. *Vanishing coefficients*
$$\sum_{t=1}^{\infty}(1-\lambda_t) = \infty, \text{ which implies } \prod_{t=1}^{\infty}\lambda_t = 0.$$
2. *Mean contraction*
$$||\mathbb{E}[\mathbf{z}_t|\mathcal{F}_{t-1}]|| \leq \lambda_t \left\|\mathbf{z}_{t-1}\right\|, a.s..$$
3. *Bounded deviation*
$$\sup\|\epsilon_t\| \to 0, \quad \sum_{t=1}^{\infty}\mathbb{E}[\|\epsilon_t\|^2] < \infty.$$

In particular, a multidimensional stochastic contraction exhibits the following behavior

1. *Contraction*
$$\mathbf{z}_t \xrightarrow{a.s.} 0.$$
2. *Kolmogorov inequality*
$$P\left(\sup_{t\geq T}\|\mathbf{z}_t\| \leq \|\mathbf{z}_T\| + \delta\right) \geq 1 - \frac{4\sqrt{d}\sum_{t=T+1}^{\infty}\mathbb{E}[\epsilon_t^2]}{\min\{\delta^2, \beta^2\}} \tag{4}$$
holds for all $T, \delta > 0$ s.t. $\beta = \|\mathbf{z}_T\| + \delta - \sqrt{d}\sup_{t>T}\|\epsilon_t\| > 0$.

In the following proof we benefit from the a.s. convergence of the difference vector to 0. And the Kolmogorov inequality gives us a PAC argument in 5.

C.2 Subsampling

We will also introduce a lemma to regularize the expected tree kernel in a finite sample case with respect to subsampling rate.

Lemma 2. *Considering a subsampled regression tree. Assume that each leaf contains no fewer than $n^{\frac{1}{d+2}}$ sample points before subsampling. If we are subsampling at rate at least $\xi = n^{-\frac{1}{d+2}\log n}$, then the expected structure vector's norm follows the rate below:*

$$\|\boldsymbol{k}_n\|_1 = 1 - O\left(\frac{1}{n}\right)$$

Proof. By Zhou and Hooker (2022), we know that $\|\mathbb{E}_{q,w}[\mathbf{S}_n]\| \leq 1$. By this we know that at least $\|\boldsymbol{k}_n\|_1 \leq 1$. The task remained is to give the distance between $\|\boldsymbol{k}_n\|_1$ and 1.

Define the subsampling index set as w. To see clearly why a rate containing n would exists, recall the definition of a structure vector with subsampling,for any $x \in A_j$,

$$\boldsymbol{s}_{n,k}(\mathbf{x}) = \boldsymbol{s}_{n,k}(x; w) = \frac{\mathbb{1}(\mathbf{x}_k \in A_j)\mathbb{1}(k \in w)}{\sum_{i=1}^{n}\mathbb{1}(\mathbf{x}_i \in A_j)\mathbb{1}(i \in w)} = \frac{\mathbb{1}(\mathbf{x}_k \in A_j)\mathbb{1}(k \in w)}{\sum_{\mathbf{x}_i \in A_j}\mathbb{1}(i \in w)}.$$

The structure vector is given by

$$\boldsymbol{s}_n(x) = [\boldsymbol{s}_{n,1}(x), \boldsymbol{s}_{n,2}(x), \cdots, \boldsymbol{s}_{n,n}(x)]^T$$

Taking expectation we have the kernel vector

$$\boldsymbol{k}_n^{\top} = \left[\frac{\mathbb{P}(\mathbf{x}_k \in A_j)\mathbb{P}(k \in w)}{\sum_{\mathbf{x}_i \in A_j}\mathbb{P}(i \in w)}\right]_{i,j}$$

Notice the expectation is taken with respect to both structure distribution and the subsampling distribution. And this vector is not defined if one leaf is empty, since the denominator would be 0. Let

the indicator I_A denote whether the leaf of interest is empty. It values on 0 if the leaf is empty and it values on 1 if the value is not empty. The L_1 norm is given by:

$$\begin{aligned}\|\boldsymbol{k}_n\|_1 &= \sum_{k=1}^{n} \mathbb{E}_{q,w}[\boldsymbol{s}_{n,k}(x)] \\ &= \sum_{k=1}^{n} (\mathbb{E}_{q,w}[\boldsymbol{s}_{n,k}(x)|\mathbb{1}_A = 0]\mathbb{P}(\mathbb{1}_A = 0) + \mathbb{E}_{q,w}[\boldsymbol{s}_{n,k}(x)|\mathbb{1}_A = 1]\mathbb{P}(\mathbb{1}_A = 1))\end{aligned}$$

If the leaf is empty, then we will have to define $\mathbb{E}_{q,w}[\boldsymbol{s}_{n,k}(x)|\mathbb{1}_A = 0] = 0$, otherwise the denominator would be 0 and the whole structure vector is undefined. If the leaf is not empty, we would then want to define the conditional expectation such that it sums up to 1:

$$\sum_{k=1}^{n} \mathbb{E}[s_{n,k}(x)|I_A = 1] = 1$$

Hence the L_1 norm above simplifies to:

$$\|\boldsymbol{k}_n\|_1 = \sum_{k=1}^{n} \mathbb{E}[\boldsymbol{s}_{n,k}(x)|\mathbb{1}_A = 1]\mathbb{P}(\mathbb{1}_A = 1) = 1 - \mathbb{P}(\mathbb{1}_A = 0)$$

Hence the problem is reduced to finding the probability of missing all subsampled points in a leaf of interest.

By assertion, the subsample size should be $\xi n = n^{\frac{d+1}{d+2}} \log n$. Consider the probability that one leaf misses all sample points $p(n, \xi)$. That is, all the subsampled points sampled are points that are in this particular leaf. Thus we are only choosing with choice $\binom{n-n^{\frac{1}{d+2}}}{\theta n}$.

$$\begin{aligned}\mathbb{P}(n,\xi) &= \frac{\binom{n-n^{\frac{1}{d+2}}}{\xi n}}{\binom{n}{\xi n}} \\ &= \frac{(n-\xi n)(n-\xi n-1)(n-\xi n-2)\cdots(n-\xi n-n^{\frac{1}{d+2}}+1)}{n(n-1)(n-2)\cdots(n-n^{\frac{1}{d+2}}+1)} \\ &\leq \left(\frac{n-\xi n}{n-n^{\frac{1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} \\ &= \left(\frac{1-\xi}{1-n^{-\frac{d+1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} \\ &= \left(\frac{1-n^{-\frac{1}{d+2}}\log n}{1-n^{-\frac{d+1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} \\ &= \left(\frac{1}{1-n^{-\frac{d+1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} \cdot \left(1-n^{-\frac{1}{d+2}}\log n\right)^{n^{\frac{1}{d+2}}}\end{aligned}$$

Notice that

$$\begin{aligned}\left(\frac{1}{1-n^{-\frac{d+1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} &\leq \left(\frac{1+n^{-\frac{d+1}{d+2}}}{1-n^{-\frac{d+1}{d+2}}+n^{-\frac{d+1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} \\ &= \left(1+n^{-\frac{d+1}{d+2}}\right)^{n^{\frac{1}{d+2}}} \\ &\leq \left(1+n^{-\frac{1}{d+2}}\right)^{n^{\frac{1}{d+2}}}\end{aligned}$$

$$\leq e$$

And meanwhile we recognize

$$\begin{aligned}\left(1-n^{-\frac{1}{d+2}}\log n\right)^{n^{\frac{1}{d+2}}} &= \left[\left(1-n^{-\frac{1}{d+2}}\log n\right)^{n^{\frac{1}{d+2}}\frac{1}{\log n}}\right]^{\log n}\\ &\leq \left(\frac{1}{e}\right)^{\log n}\\ &= \frac{1}{e^{\log n}}\\ &= \frac{1}{n}\end{aligned}$$

Hence we have

$$\begin{aligned}&\left(\frac{1}{1-n^{-\frac{d+1}{d+2}}}\right)^{n^{\frac{1}{d+2}}}\cdot\left(1-n^{-\frac{1}{d+2}}\log n\right)^{n^{\frac{1}{d+2}}}\\ &\leq e\cdot\left(1-n^{-\frac{1}{d+2}}\log n\right)^{n^{\frac{1}{d+2}}}\\ &= O\left(\frac{1}{n}\right)\end{aligned}$$

Hence the probability of a single leaf missing all subsampled points is of rate $O\left(\frac{1}{n}\right)$. And hence $\|\boldsymbol{k}_n\|_1 = 1 - O\left(\frac{1}{n}\right)$. □

C.3 Proof for Theorem 1

Combining all the results above, the proof for Theorem 1 is as follow.

Proof. To show Algorithm 1 is a stochastic contraction mapping, we aim to show the conditions 1-3 are satisfied. Define the difference $\mathbf{z}_t := \widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}^*$. And define the partial ensemble $\widetilde{\mathbf{y}}_b := \frac{1}{b}\sum_{k=1}^{b} X_k\widehat{t}_k, X_k \overset{i.i.d.}{\sim} \text{Ber}(q)$. To check mean contraction, first notice as $b\to\infty$, $\text{tr}(\text{Var}(\widetilde{\mathbf{y}}_b)) \leq \frac{q(1-q)nM}{b} \to 0$. Hence with b large enough, we can get rid of the truncation.

$$\begin{aligned}\|\mathbb{E}[\mathbf{z}_b|\mathcal{F}_{b-1}]\| &= \left\|\mathbb{E}\left[\frac{b-1}{b}\widehat{\mathbf{y}}_{b-1} + \frac{\lambda}{b}\mathbf{S}_n(\mathbf{y}-\Gamma_M(\widetilde{\mathbf{y}}_{b-1})) - \widehat{\mathbf{y}}^*\middle|\mathcal{F}_{b-1}\right]\right\|\\ &= \left\|\frac{b-1}{b}(\widehat{\mathbf{y}}_{b-1}-\widehat{\mathbf{y}}^*) + \frac{\lambda}{b}\mathbb{E}[\mathbf{S}_n](\mathbf{y}-\mathbb{E}[\Gamma_M(\widetilde{\mathbf{y}}_{b-1})|\mathcal{F}_{b-1}]) - \frac{1}{b}\widehat{\mathbf{y}}^*\right\|\\ &\leq \frac{b-1}{b}\|\widehat{\mathbf{y}}_{b-1}-\widehat{\mathbf{y}}^*\| + \left\|\frac{\lambda}{b}\mathbb{E}[\mathbf{S}_n](\mathbf{y}-\mathbb{E}[\Gamma_M(\widetilde{\mathbf{y}}_{b-1})|\mathcal{F}_{b-1}]) - \frac{\lambda}{b}\mathbb{E}[\mathbf{S}_n](\mathbf{y}-q\widehat{\mathbf{y}}^*)\right\|\\ &= \frac{b-1}{b}\|\mathbf{z}_{b-1}\| + \left\|\frac{\lambda}{b}\mathbb{E}[\mathbf{S}_n](q\widehat{\mathbf{y}}^*-\mathbb{E}[\Gamma_M(\widetilde{\mathbf{y}}_{b-1})|\mathcal{F}_{b-1}])\right\|\\ &= \frac{b-1}{b}\|\mathbf{z}_{b-1}\| + \left\|\frac{\lambda}{b}\mathbb{E}[\mathbf{S}_n](q\widehat{\mathbf{y}}^*-\mathbb{E}[\widetilde{\mathbf{y}}_{b-1}|\mathcal{F}_{b-1}])\right\|\\ &\leq \frac{b-1+\lambda q}{b}\|\widehat{\mathbf{y}}_{b-1}-\widehat{\mathbf{y}}^*\| \leq \|\widehat{\mathbf{y}}_{b-1}-\widehat{\mathbf{y}}^*\|\end{aligned}$$

Next we check for bounded deviations.

$$\begin{aligned}\|\epsilon_b\| &= \|\mathbf{z}_b - \mathbb{E}[\mathbf{z}_b|\mathcal{F}_{b-1}]\|\\ &= \|(\widehat{\mathbf{y}}_b-\widehat{\mathbf{y}}^*) - (\mathbb{E}[\widehat{\mathbf{y}}_b-\widehat{\mathbf{y}}^*|\mathcal{F}_{b-1}])\|\\ &= \left\|\widehat{Y}_b - \mathbb{E}[\widehat{Y}_b|\mathcal{F}_{b-1}]\right\|\\ &= \left\|\frac{\lambda}{b}(\mathbb{E}[\mathbf{S}_n]-\mathbf{S}_n)(\mathbf{y}-\Gamma_M(\widetilde{\mathbf{y}}_{b-1}) + \mathbb{E}[\Gamma_M(\widetilde{\mathbf{y}}_{b-1})|\mathcal{F}_{b-1}])\right\|\end{aligned}$$

$$\leq \frac{\lambda}{b} \|\mathbb{E}[\mathbf{S}_n] - \mathbf{S}_n\| \, \|Y - \Gamma_M(\widetilde{\mathbf{y}}_{b-1}) + \mathbb{E}[\Gamma_M(\widetilde{\mathbf{y}}_{b-1})|\mathcal{F}_{b-1}]\|$$

Now, by Zhou and Hooker (2022), we know $\mathbb{E}[\mathbf{S}_n]$ has both row sum and column sum no greater than 1, hence we see that$\|\mathbb{E}[\mathbf{S}_n]\| \leq \sqrt{\|\mathbb{E}[\mathbf{S}_n]\|_1 \cdot \|\mathbb{E}[\mathbf{S}_n]\|_\infty} \leq \sqrt{1 \times n} = \sqrt{n}$. Then, it can be seen that

$$\|\epsilon_b\| \leq \frac{\lambda}{b} \cdot (1 + \sqrt{n}) \cdot 2M$$

Then we can show that $\sum_{b=1}^{\infty} \|\epsilon_b\|^2 = \sum_{b=1}^{\infty} O(\frac{1}{b^2}) < \infty$.

Similarly we can show both conditions for Algorithm 2. Replacing the stepwise difference according to the new update rule gives:

$$\begin{aligned}
\|\mathbb{E}[\mathbf{z}_{b+1} \mid \mathcal{F}_b]\| &= \|\mathbb{E}[\widehat{\mathbf{y}}_{b+1,K} - \widehat{\mathbf{y}}_K^* \mid \mathcal{F}_b]\| \\
&= \left\| \frac{b}{b+1} \widehat{\mathbf{y}}_{b,K} + \frac{1}{b+1} \mathbb{E}[\mathbf{S}_n] \sum_{k=1}^{K} (\mathbf{y} - \widetilde{\mathbf{y}}_{b,k-1}) - \widehat{\mathbf{y}}_K^* \right\| \\
&= \left\| \frac{b}{b+1} \widehat{\mathbf{y}}_{b,K} + \frac{1}{b+1} \mathbb{E}[\mathbf{S}_n] \sum_{k=1}^{K} \Big(\mathbf{y} - \big(\widehat{\mathbf{y}}_{b,K} - \Gamma_M\big(\tfrac{1}{b} \sum_{g=1}^{b} \widehat{\mathbf{t}}_{g,k}\big)\big)\Big) - \widehat{\mathbf{y}}_K^* \right\| \\
&= \left\| \frac{b}{b+1} \widehat{\mathbf{y}}_{b,K} + \frac{K}{b+1} \mathbb{E}[\mathbf{S}_n] \, \mathbf{y} - \frac{K}{b+1} \mathbb{E}[\mathbf{S}_n] \, \widehat{\mathbf{y}}_{b,K} + \frac{1}{b+1} \mathbb{E}[\mathbf{S}_n] \sum_{k=1}^{K} \Gamma_M\big(\tfrac{1}{b} \sum_{g=1}^{b} \widehat{\mathbf{t}}_{g,k}\big) - \widehat{\mathbf{y}}_K^* \right\| \\
&= \left\| \frac{1}{b+1} \big(bI - K \, \mathbb{E}[\mathbf{S}_n]\big) \big(\widehat{\mathbf{y}}_{b,K} - \widehat{\mathbf{y}}_K^*\big) + \frac{1}{b+1} \mathbb{E}[\mathbf{S}_n] \Big(\sum_{k=1}^{K} \Gamma_M\big(\tfrac{1}{b} \sum_{g=1}^{b} \widehat{\mathbf{t}}_{g,k}\big) - \widehat{\mathbf{y}}_K^* \Big) \right\| \\
&\leq \frac{1}{b+1} \, \|b\mathbf{I} - K \, \mathbb{E}[\mathbf{S}_n]\| \, \|\widehat{\mathbf{y}}_{b,K} - \widehat{\mathbf{y}}_K^*\| + \frac{1}{b+1} \, \|\mathbb{E}[\mathbf{S}_n]\| \left\| \sum_{k=1}^{K} \Gamma_M\big(\tfrac{1}{b} \sum_{g=1}^{b} \widehat{\mathbf{t}}_{g,k}\big) - \widehat{\mathbf{y}}_K^* \right\| \\
&\leq \frac{1}{b+1} \, \|b\mathbf{I} - K \, \mathbb{E}[\mathbf{S}_n]\| \, \|\widehat{\mathbf{y}}_{b,K} - \widehat{\mathbf{y}}_K^*\| + \frac{1}{b+1} \, \|\mathbb{E}[\mathbf{S}_n]\| \left\| \Gamma_{KM}\big(\sum_{k=1}^{K} \tfrac{1}{b} \sum_{g=1}^{b} \widehat{\mathbf{t}}_{g,k}\big) - \widehat{\mathbf{y}}_K^* \right\| \\
&\leq \frac{1}{b+1} \left(\|b\mathbf{I} - K \, \mathbb{E}[\mathbf{S}_n]\| + \|\mathbb{E}[\mathbf{S}_n]\| \right) \|\widehat{\mathbf{y}}_{b,K} - \widehat{\mathbf{y}}_K^*\| \\
&\leq \frac{1}{b+1} \left(b + 1 - O(\frac{1}{n}) \right) \|\widehat{\mathbf{y}}_{b,K} - \widehat{\mathbf{y}}_K^*\| \\
&< \|\mathbf{z}_b\|
\end{aligned}$$

Hence we have checked the mean contraction condition. Checking the bounded deviation condition works in a similar way for Algorithm 2 since it shares the same expected tree kernel with Algorithm 1 and the tree signal is also bounded by M. Hence the stochastic contraction mapping follows. □

D Conditional CLT Around KRR Predictions

Doing so requires control on the weighting vectors $\boldsymbol{r}_n(x)$. And we will begin such analysis on giving the rate of the expected structure vector $\boldsymbol{k}_n(x)$ first. This is because recall that in both algorithms our final weight vector $\boldsymbol{r}_n(x)$ is a linear map of the voting vector $\boldsymbol{k}_n(x)$.To begin with, we will first notice that Lemma 2 implies Lemma 3.

D.1 Bounding the weights

Lemma 3. *Suppose the assumptions for Lemma 2 are satisfied. We have* $\left| \sum_{i=1}^{n} \boldsymbol{r}_{n,i}^{D} - \frac{\lambda}{1+\lambda q} \right| = O\left(\frac{1}{n}\right), \quad \left| \sum_{i=1}^{n} \boldsymbol{r}_{n,i}^{P} - 1 \right| = O\left(\frac{1}{n}\right).$

Proof. We begin with the analysis on Algorithm 1. Consider the expansion

$$\left[\frac{1}{\lambda} I + q\mathbf{K}_n \right]^{-1} = \lambda \sum_{i=0}^{\infty} \left((\lambda q)^{2i} \mathbf{K}_n^{2i} - (\lambda q)^{2i+1} \mathbf{K}_n^{2i+1} \right).$$

We examine the column sums of each of the matrix powers. Start with K_n^2,

$$\sum_{i=1}(\mathbf{K}_n^2)_{i,1} = \sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{K}_n)_{i,j}(\mathbf{K}_n)_{j,1} = \sum_{j=1}^{n}(\mathbf{K}_n)_{j,1}\sum_{i=1}^{n}(\mathbf{K}_n)_{i,j}.$$

Take subsampling in to consideration, since K_n consists of structure vectors of sample points, hence for some $c > 0$, we can bound the row sum:

$$1 - \frac{c}{n} \le \sum_{i=1}^{n}(\mathbf{K}_n)_{i,j} \le 1, \quad i = 1, \dots, n.$$

which also hold true for

$$1 - \frac{c}{n} \le \sum_{j=1}^{n}(\mathbf{K}_n)_{j,1} \le 1, \quad i = 1, \dots, n.$$

Given K_n is nonnegative, multiply the inequalities above and notice it is actually the row sum of K_n^2, we have

$$\left(1 - \frac{c}{n}\right)^2 \le \sum_{i=1}(\mathbf{K}_n^2)_{i,1} = \sum_{j=1}^{n}(\mathbf{K}_n)_{j,1}\sum_{i=1}^{n}(\mathbf{K}_n)_{i,j} \le 1.$$

Repeating the same procedure above yields

$$\left(1 - \frac{c}{n}\right)^m \le \sum_{i=1}(\mathbf{K}_n^m)_{i,1} \le 1.$$

Therefore,

$$\begin{aligned}
\lambda\left(\frac{1}{1-\lambda^2q^2(1-\frac{c}{n})^2} - \frac{\lambda q}{1-\lambda^2q^2}\right) &\le \sum_{j=1}^{n}\left[\frac{1}{\lambda}I + q\mathbf{K}_n\right]^{-1}_{j,1} \\
&= \lambda\left(\sum_{i=0}^{\infty}(\lambda q)^{2i}(\mathbf{K}_n^{2i})_{j,1} - (\lambda q)^{2i+1}(\mathbf{K}_n^{2i+1})_{j,1}\right) \\
&\le \lambda\left(\frac{1}{1-\lambda^2q^2} - \frac{\lambda q}{1-\lambda^2q^2(1-\frac{c}{n})^2}\right),
\end{aligned}$$

where both the LHS and RHS reduce to $\frac{\lambda}{1+\lambda q} + O\left(\frac{1}{n}\right)$. So is true for any column sum of $\left[\frac{1}{\lambda}I + q\mathbf{K}_n\right]^{-1}$.And we know that $\boldsymbol{k}_n$ is nonnegative and $1 - \|\boldsymbol{k}_n\|_1 = O\left(\frac{1}{n}\right)$ and $\boldsymbol{k}_n$ simply reweights the columns. Hence we prove the convergence of weights $\sum_{i=1}^{n}\boldsymbol{r}_{n,i}$ for Algorithm 1.

The analysis for Algorithm 2 is similar. We simply substitute $\lambda = K, q = \frac{K-1}{K}$. This will give the bound:

$$\begin{aligned}
K\left(\frac{1}{1-(K-1)^2(1-\frac{c}{n})^2} - \frac{K-1}{1-(K-1)^2}\right) &\le \sum_{j=1}^{n}\left[\frac{1}{K}I + \frac{K-1}{K}\mathbf{K}_n\right]^{-1}_{j,1} \\
&= K\left(\sum_{i=0}^{\infty}(K-1)^{2i}(\mathbf{K}_n^{2i})_{j,1} - (K-1)^{2i+1}(\mathbf{K}_n^{2i+1})_{j,1}\right) \\
&\le K\left(\frac{1}{1-(K-1)^2} - \frac{K-1}{1-(K-1)^2(1-\frac{c}{n})^2}\right),
\end{aligned}$$

We can reduce both sides of the inequalities to $1 + O\left(\frac{1}{n}\right)$ similarly. That proves the converging weight of Algorithm 2. □

Intuitively, knowing that the weights sum to (almost) a constant tells us the total "mass" of our weighting vector is fixed. If we have leaf size assumptions such that no single weight can be too large, the mass must be spread out over many small pieces. Whenever you distribute a fixed amount of weight across many entries, the Euclidean length of the vector necessarily shrinks. In other words, a near-unit ℓ_1 norm plus a bound on the largest coordinate forces the ℓ_2 norm to be small. We control the rate of $\boldsymbol{r}_n^{D,P}$ in Lemma 1. To do so, we need to both upper bound and lower bound $\|\boldsymbol{k}_n\|$, which is helped by Assumption 5, implying $\inf_{A\in q\in Q_n}\sum_{i=1}^{n}\mathbb{1}(\mathbf{x}_i \in A) = \Omega\left(n^{\frac{1}{d+1}}\right)$.

D.2 Proof for Lemma 1

We will make use of the $\Omega\left(n^{\frac{1}{d+1}}\right)$ in Assumption 7 to control $\|\boldsymbol{k}_n\|$ and $\|\boldsymbol{r}_n\|$. The first bound in the $\max$ operator is vital for our unconditional CLT analysis Lemma 7. We will first show Lemma 1 which is sufficient for us to get a conditional CLT.

Proof. To bound $\|\boldsymbol{k}_n\|$, recall:

$$\boldsymbol{k}_{nj} = \mathbb{E}[\boldsymbol{s}_{n,j}(x)] = \mathbb{E}\left[\frac{\mathbb{1}(x_j \in A)}{\sum_{j=1}^{n} \mathbb{1}(x_j \in A)}\right], x \in A$$

encodes the expected influence of point x_j on a point of interest x among other n points. Then the condition

$$\inf_{A \in q \in Q_n} \sum_{i=1}^{n} \mathbb{1}(x_i \in A) \geq \Omega\left(n^{\frac{1}{d+1}}\right)$$

implies that $\boldsymbol{k}_{nj} = O\left(n^{-\frac{1}{d+1}}\right)$, since we can't have total weights of more than $O(1)$ in a leaf. By Lemma 2, $\|\boldsymbol{k}_n\|_1 \leq 1$,

$$\|\boldsymbol{k}_n\| \leq \sqrt{\|\boldsymbol{k}_n\|_1 \|\boldsymbol{k}_n\|_\infty} = O\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right).$$

By assetion $|B_n| = \Omega\left(n \cdot d_n^d\right)$, there are at most

$$\Omega\left(n \cdot d_n^d\right) = \Omega\left(n^{\frac{1}{d+1}}\right)$$

k_{nj}'s that are positive. Equivalently, we can have the magnitude of each non-zero $\boldsymbol{k}_{nj}$ is lower bounded by $\Omega\left(n^{-\frac{1}{d+1}}\right)$. This holds also because the total weights can't be larger than $O(1)$. Since $\|\boldsymbol{k}_n\|_1 = 1 - O(n^{-1})$,

$$\|\boldsymbol{k}_n\| = \Omega\left(\sqrt{\left(n^{-\frac{1}{d+1}}\right)^2 \cdot n^{\frac{1}{d+1}}}\right) = \Omega\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right).$$

To check the rate of $\|\boldsymbol{r}_n\|$, since r_n is a mapped from k_n by the KRR matrix. For Algorithm 1 it's $[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n]$ and for Algorithm 2 is $[\frac{1}{K}I + \frac{K-1}{K}K_n]^{-1}$ and we know

$$\frac{\lambda}{1+\lambda q} \leq eigen\left(\left[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n\right]^{-1}\right) \leq \lambda, \quad 1 \leq eigen\left(\left[\frac{1}{K}\mathbf{I} + \frac{K-1}{K}\mathbf{K}_n\right]^{-1}\right) \leq K$$

Since we can fix K to be a constant, $\|\boldsymbol{r}_n\|$ will enjoy similar rates as $\|\boldsymbol{k}_n\|$. □

This rate complies with the Lindeberg-Feller conditions, hence we can establish the conditional CLT.

D.3 Conditional Asymptotic Normality on Training Set

Theorem 4 (Conditional Asymptotic Normality for BRAT-D and BRAT-P Predictions)**.** *For any $\mathbf{x} \in [0,1]^d$, write $f(\mathbf{X}_n) = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))^\top$. Then under Assumptions 1, 2, 4 and assumptions in Lemma 1, we have*

$$\frac{\widehat{\boldsymbol{f}}_n^{D,P}(\mathbf{x}) - \langle \boldsymbol{r}_n^{D,P}, \boldsymbol{f}(\mathbf{X}_n)\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \xrightarrow{d} \mathcal{N}(0, \sigma^2).$$

Proof. For notational convenience, since $\|\boldsymbol{r}_n\|$ yielded by both Algorithm 1 and Algorithm 2 shares the same rate in n, we will suppress the notation of D, P in this proof, and denote the kernel ridge regression matrix respectively with KRR^D and KRR^P. Also define $c^D = \frac{\lambda}{1+\lambda q}, c^P = 1$.

Write

$$\widehat{\boldsymbol{f}}_n(x) - \boldsymbol{r}_n^T \boldsymbol{f}(X_n) = \boldsymbol{r}_n^T \vec{\epsilon}_n.$$

To establish the CLT, we verify the Lindeberg–Feller condition for the triangular array $\boldsymbol{r}_n^T \vec{\epsilon}_n$, i.e. for any $\delta > 0$,

$$\lim_n \frac{1}{\|\boldsymbol{r}_n\|^2 \sigma^2} \sum_{i=1}^n \mathbb{E}\left[(\boldsymbol{r}_{ni}\epsilon_i)^2 \mathbb{1}(|\boldsymbol{r}_{ni}\epsilon_i| > \delta \|\boldsymbol{r}_n\| \sigma_)\right] = 0.$$

Since $\|\boldsymbol{k}_n\|_\infty = O\left(n^{-\frac{1}{d+2}}\right)$ and $\mathsf{KRR}^{D,P}$ having row sums of $C^{D,P} + O\left(n^{-1}\right)$, we have

$$\|\boldsymbol{r}_n\|_\infty \leq \|\boldsymbol{k}_n\|_\infty \cdot \left\|\mathsf{KRR}^{D,P}\right\|_1 = O\left(n^{-\frac{1}{d+1}}\right).$$

Furthermore, since $\|\boldsymbol{r}_n\| = \Theta(n^{-\frac{1}{2}\frac{1}{d+1}})$, we get

$$\frac{\|\boldsymbol{r}_n\|_\infty}{\|\boldsymbol{r}_n\|} = O\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right),$$

This allows us to check out the Lindeberg-Feller conditions:

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E}\left[(\boldsymbol{r}_{ni}\epsilon_i)^2 \mathbb{1}(|\boldsymbol{r}_{ni}\epsilon_i| > \delta \|\boldsymbol{r}_n\| \sigma)\right] &\leq \sum_{i=1}^n \boldsymbol{r}_{ni}^2 \sqrt{\mathbb{E}[\epsilon_i^4] \cdot \mathbb{E}[\mathbb{1}(|\boldsymbol{r}_{ni}\epsilon_i| > \delta \|\boldsymbol{r}_n\| \sigma^2]} \\
&\leq \sum_{i=1}^n \boldsymbol{r}_{ni}^2 \sqrt{\mathbb{E}[\epsilon_i^4]} \cdot \sqrt{P\left(|\epsilon_i| \geq \frac{\delta \|\boldsymbol{r}_n\| \sigma}{\boldsymbol{r}_{ni}}\right)} \\
&\leq \sum_{i=1}^n \boldsymbol{r}_{ni}^2 \sqrt{\mathbb{E}[\epsilon_i^4]} \sqrt{2\exp\left(-\frac{1}{2\sigma^2} \cdot \left(\frac{\delta \|\boldsymbol{r}_n\| \sigma}{\boldsymbol{r}_{ni}}\right)^2\right)} \\
&\leq \|\boldsymbol{r}_n\|^2 \exp\left(-O\left(n^{\frac{1}{d+1}}\right)\right) \longrightarrow 0,
\end{aligned}$$

Since ϵ is sub-Gaussian noise, the concentration bound holds by definition of ϵ. This concludes the proof for Theorem 4. □

E Extension to Random Design and Exponential Locality

Now we move on to the discussions in which our input to the model is no longer restricted only to the training set $\mathbf{X}_n$, but to any $\mathbf{x} \in [0,1]^d$. To do so, we begin by defining a new probability space and, of course, a new probability measure. This can be uniquely designed by a Kolmogorov's extension theorem.

Write the coordinate projection $\Pi_n = (\pi_n(\mathbf{X}), \pi_n(\vec{\epsilon}))$ as the finite-dimensional projection containing the first n samples. Under this framework, we can obtain the corresponding expected structure vector $\boldsymbol{k}_n$, expected structure matrix (kernel matrix) K_n and the standardized prediction error $\rho_n(\mathbf{X}, \vec{\epsilon})$, given by

$$\rho_n(\mathbf{X}, \epsilon) = \frac{\widehat{\boldsymbol{f}}_n^{D,P}(\mathbf{x}; \Pi_n) - \langle \boldsymbol{k}_n^\top(\mathbf{x}; \Pi_n) \mathsf{KRR}^{D,P}(\Pi_n), \boldsymbol{f}(\Pi_n)\rangle}{\left\|\boldsymbol{k}_n(\mathbf{x}; \Pi_n)^\top \mathsf{KRR}^{D,P} \Pi_n\right\|}$$

Here, ρ_n denotes the prediction error after using n random samples incorporating both the dropout mechanism and the sample randomness.

To prove our desired CLT under random design, we need to first introduce a lemma that allows us to propagate the normality from fixed sample to random sample cases. This is taken care of by Lemma 8, which allows us to claim our CLTs if we verify it being correct for $a.s. \forall \mathbf{x} \in [0,1]^d$. Then, we seek an almost sure validation of the assumptions of Theorem 4 under the random sample sequence $\Pi_n = (\pi_n(\mathbf{X}), \pi_n(\vec{\epsilon}))$, which is helped by our Assumption 6. Formally, it is as shown below:

Lemma 4. *Suppose we have random sample $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n \subset [0,1]^d \times \mathbb{R}$ for each n. If for any small $\alpha, \nu > 0$ s.t.*

$$|Q_n| = O\left(\frac{1}{n} \exp\left(\frac{1}{2} n^{\frac{1}{d+2}-\nu} - n^\alpha\right)\right),$$

then

$$\frac{\widehat{\boldsymbol{f}}_n^{D,P}(\mathbf{x}) - \langle \boldsymbol{r}_n^{D,P}, \boldsymbol{f}(\mathbf{X}_n)\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \xrightarrow{d} \mathcal{N}(0, \sigma^2).$$

Proof. We will direct the reader to Zhou and Hooker (2022). Despite the introduction of stochasticity in dropout, extending the CLT of 4 does not require additional effort. The intuition for the proof is as follows: By restricting the tree space, one bounds the probability of the assumptions being violated to be summable, and per Borel-Cantelli will take place almost surely. Tree space cardinality here gives a nice union probability bound to achieve that. □

We define the following notation for later use: For any n-vector v and an index set D, denote

$$v\big|_D = \begin{bmatrix} v_1 \cdot \mathbb{1}(1 \in D) \\ \vdots \\ v_n \cdot \mathbb{1}(n \in D) \end{bmatrix}.$$

This implies the decomposition that $v = v\big|_D + v\big|_{D^c}$.

We are now almost ready to prove the main theorem. However, to check Lindeberg-Feller conditions under random design, one needs to establish exponential decay of points that sit far away from the point of interest. That is, $\|\boldsymbol{r}_n|_{D^c}\|$ for a well chosen ball centered at $\mathbf{x}$ needs to vanish to 0 rather quickly. And we prove it in the following lemma. To control this for Algorithm 1, we have the following lemma:

Lemma 5. *Given sample* $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ *and a point of interest* $\mathbf{x}$*, set* $l_n = \frac{\log n}{-\log \lambda q} = c_1 \log n, c_1 = \log \frac{1}{\lambda q}$ *and define index set* $D_n = \{i : |\mathbf{x}_i - \mathbf{x}| \le l_n \cdot d_n\}$*, then*

$$\left\|\boldsymbol{r}_n^D\big|_{D_n^c}\right\|_1 = O\left(\frac{1}{n}\right).$$

Proof. Let's first write out the expression of the desired peripheral points:

$$\left\|\boldsymbol{r}_n^D\big|_{D_n^c}\right\|_1 = \sum_{|\mathbf{x}-\mathbf{x}_i| > l_n \cdot d_n} |\boldsymbol{r}_{ni}^D|$$

Recall:

$$\boldsymbol{k}_n^T = \mathbb{E}[\boldsymbol{s}_n(\mathbf{x})] \to \boldsymbol{k}_{nj} = \mathbb{E}[\boldsymbol{s}_{n,j}(\mathbf{x})] = \mathbb{E}[\frac{\mathbb{1}(\mathbf{x}_j \in A)}{\sum_{j=1}^n \mathbb{1}(\mathbf{x}_j \in A)}], \mathbf{x} \in A$$

This term would be zero if two points are not even in the same leaf, i.e. $|\mathbf{x} - \mathbf{x}_j| > d_n$. Hence the sum inside would be simplified to

$$\sum_{|\mathbf{x}-\mathbf{x}_i| > l_n \cdot d_n} \left| \sum_{|\mathbf{x}_j - \mathbf{x}| \le d_n} \boldsymbol{k}_{nj} \left[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n\right]^{-1}_{j,i} \right|$$

And also notice, by decreasing $l_n d_n$ to $(l_n - 1)d_n$, we are actually allowing more peripheral points outside the local region D_n and hence we have the inequality at line 4 below. Completely. it is given as:

$$\begin{aligned}
\left\|\boldsymbol{r}_n^D\big|_{D_n^c}\right\|_1 = \sum_{|\mathbf{x}-\mathbf{x}_i| > l_n \cdot d_n} |\boldsymbol{r}_{ni}^D| &= \sum_{|\mathbf{x}-\mathbf{x}_i| > l_n \cdot d_n} \left| \sum_j \boldsymbol{k}_{nj} \left[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n\right]^{-1}_{j,i} \right| \\
&= \sum_{|\mathbf{x}-\mathbf{x}_i| > l_n \cdot d_n} \left| \sum_{|\mathbf{x}-\mathbf{x}_j| \le d_n} \boldsymbol{k}_{nj} \left[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n\right]^{-1}_{j,i} \right| \\
&\le \sum_{|\mathbf{x}-\mathbf{x}_j| \le d_n} \boldsymbol{k}_{nj} \sum_{|\mathbf{x}-\mathbf{x}_i| > l_n \cdot d_n} \left| \left[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n\right]^{-1}_{j,i} \right|
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} \sum_{|\mathbf{x}_i-\mathbf{x}_j|>(l_n-1)\cdot d_n} \left| \left[\frac{1}{\lambda}\mathbf{I} + q\mathbf{K}_n\right]^{-1}_{j,i} \right| \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} \sum_{|\mathbf{x}_i-\mathbf{x}_j|>(l_n-1)\cdot d_n} \lambda \sum_{l=l_n}^{\infty} (\lambda q)^l [\mathbf{K}_n^l]_{j,i} \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} \sum_{l=l_n}^{\infty} (\lambda q)^{l+1} \\
&\leq \lambda \sum_{l=l_n}^{\infty} (\lambda q)^l = \frac{\lambda}{1-\lambda q}\frac{1}{n}.
\end{aligned}
$$

The lower bound in the sum of the powers of $\mathbf{K}_n$ follows this argument. To compute higher powers of $\mathbf{K}_n$, consider how entries of K_n^l are calculated:

$$
\mathbf{K}_n^l[i,j] = \sum_{k_1,k_2,\ldots,k_{l-1}} \mathbf{K}_n[i,k_1]\mathbf{K}_n[k_1,k_2]\ldots\mathbf{K}_n[k_{l-1},j].
$$

For $\mathbf{K}_n^l[i,j] \neq 0$, there must exist a chain of intermediate points $\{k_1, k_2, \ldots, k_{l-1}\}$ such that:

$$
|\mathbf{x}_i - \mathbf{x}_{k_1}| \leq d_n, \quad |\mathbf{x}_{k_1} - \mathbf{x}_{k_2}| \leq d_n, \quad \ldots, \quad |\mathbf{x}_{k_{l-1}} - \mathbf{x}_j| \leq d_n.
$$

By summing the distances and by a triangular inequality, the locality condition propagates:

$$
|\mathbf{x}_i - \mathbf{x}_j| \leq l \cdot d_n,
$$

where l is the number of steps (or multiplications) required to connect x_i and x_j through the intermediate points. Now notice the summing condition $|x_i - x_j| > (l_n - 1)d_n$ requires us to have such chain that is longer than $l_n - 1$ at least. Hence the kernel's power $\mathbf{K}_n^l$ would be 0 for any $l \leq l_n - 1$, since such chain doesn't exist at all. □

To control weights from distant points for Algorithm 2 is a bit more complicated. We need to make use of Assumption 7. This assumption is inspired by the definition in Athey et al. (2018), where we state below.

Definition 3 (α-regular tree predictor)**.** *A tree predictor grown by recursive partitioning is called* α-regular *for some $\alpha > 0$ if either*

1. ***(Standard case)** each split leaves at least a fraction α of the available training examples on* each *side of the split, and furthermore the trees are fully grown to depth k for some $k \in \mathbb{N}$. Equivalently, every terminal node contains between k and $2k-1$ observations;*
2. ***(SVI case)** It satisfies part (a) when applied to the sample used to fit the leaf value. In this case this will be our response vector Y.*

Lemma 6. *If a tree is $\frac{1}{2}$-regular with the terminal leaf size being lower bounded by k, where k is a constant, then the operator norm of $\mathbf{P}$ is $O(1)$.*

Proof. Define $\alpha_* = \max\{\alpha, 1-\alpha\}$. By definition of α-regularity, the probability of point i, j falling into the same leaf after one split would be upper bounded by α_*. And since we have the minimal terminal leaf size requirement, for any leaf with depth d, it should comply with:

$$
\begin{cases}
k \leq \alpha_*^d n \leq 2k-1, \\
k \leq (1-\alpha_*)^d n \leq 2k-1.
\end{cases}
$$

which translates into $\frac{1}{\ln(1-\alpha_*)}\ln(\frac{2k-1}{n}) \leq d \leq \frac{1}{\ln(\alpha_*)}\ln(\frac{k}{n})$.

Notice that the probability of two points sharing a leaf is the probability that they survive all splits down the tree. Then we can bound the collision probability by:

$$
\begin{aligned}
p_{i,j} &\leq \alpha_*^{d_{min}} \\
&= \alpha_*^{\frac{1}{\ln(1-\alpha_*)}\ln(\frac{2k-1}{n})}
\end{aligned}
$$

$$= (\frac{2k-1}{n})^{\frac{\ln \alpha_*}{\ln(1-\alpha_*)}}$$
$$= (\frac{2k-1}{n})^{\log_{1-\alpha_*} \alpha_*}$$
$$= \frac{2k-1}{n}$$

Then $\sum_i p_{i,j} = O(1)$ □

Well-behaved leaves exclude non-decaying distant weights. Formally, we can prove the lemma below:

Lemma 7. *Given sample* $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ *and a point of interest* $\mathbf{x}$*, set* $l_n = -\frac{\ln n}{\ln(\xi/2)} = c \ln n$ *and define index set* $D_n = \{i : |\mathbf{x}_i - \mathbf{x}| \leq l_n \cdot d_n\}$*. then*

$$\left\| r_n^P \big|_{D_n^c} \right\|_1 = O\left(\frac{1}{n}\right).$$

Proof. We will use the same matrix expansion technique we used in Lemma5. By analogy,

$$\begin{aligned}
\left\| \boldsymbol{r}_n^P \big|_{D_n^c} \right\|_1 = \sum_{|\mathbf{x}-\mathbf{x}_i|>l_n \cdot d_n} |\boldsymbol{r}_{ni}^P| &= \sum_{|\mathbf{x}-\mathbf{x}_i|>l_n \cdot d_n} \left| \sum_j \boldsymbol{k}_{nj} \left[\frac{1}{K}\mathbf{I} + \frac{K-1}{K}\mathbf{K}_n \right]^{-1}_{j,i} \right| \\
&= \sum_{|\mathbf{x}-\mathbf{x}_i|>l_n \cdot d_n} \left| \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} \left[\frac{1}{K}\mathbf{I} + \frac{K-1}{K}\mathbf{K}_n \right]^{-1}_{j,i} \right| \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} \sum_{|\mathbf{x}-\mathbf{x}_i|>l_n \cdot d_n} \left| \left[\frac{1}{K}\mathbf{I} + \frac{K-1}{K}\mathbf{K}_n \right]^{-1}_{j,i} \right| \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} \sum_{|\mathbf{x}_i-\mathbf{x}_j|>(l_n-1) \cdot d_n} \left| \left[\frac{1}{K}\mathbf{I} + \frac{K-1}{K}\mathbf{K}_n \right]^{-1}_{j,i} \right| \\
&= \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} \sum_{|\mathbf{x}_i-\mathbf{x}_j|>(l_n-1) \cdot d_n} K \left| \sum_{l=l_n}^{\infty} (-1)^l (K-1)^l [\mathbf{K}_n^l]_{j,i} \right| \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} K \sum_{l\geq l_n} (K-1)^l \sum_{|\mathbf{x}_i-\mathbf{x}_j|>(l_n-1) \cdot d_n} |[\mathbf{K}_n^l]_{i,j}| \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} K \sum_{l\geq l_n} (K-1)^l \sum_{i=1}^{n} |[\mathbf{K}_n^l]_{i,j}| \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} K \sum_{l\geq l_n} (K-1)^l \left\| \mathbf{K}_n^l \right\|_\infty \\
&\leq \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} K \sum_{l\geq l_n} (K-1)^l \left\| \mathbf{K}_n \right\|_\infty^l
\end{aligned}$$

Denote the probability of any point $\mathbf{x}_i, \mathbf{x}_j$ sharing the same leaf as p_{ij}. By definition of the kernel matrix, we can write each element as:

$$\begin{aligned}
[\mathbf{K}_n]_{i,j} &= \mathbb{E}_{q,w}\Big[[\mathbf{S}_n]_{i,j} \Big] \\
&= \mathbb{E}_w\Big[[\mathbf{S}_n]_{i,j} \Big| \mathbf{x}_i, \mathbf{x}_j \text{ are in the same leaf} \Big] p_{ij} \\
&= \mathbb{E}[\frac{1}{X}] p_{i,j}
\end{aligned}$$

where $X \sim \text{Binomial}([m, M], \xi)$, with m being the minimal number of points in a leaf and M being the maximal number of points within a leaf. Since $X = m, \cdots, M$, it follows that $\frac{1}{X} \leq \frac{1}{m}$ almost

surely and hence $\mathbb{E}[\frac{1}{X}] \leq \frac{1}{m\xi}$. Then by assumption $m = \max\{\Omega\left(n^{\frac{1}{d+2}}\right), \frac{K-1}{\xi}\}$, it follows that $\mathbb{E}[\frac{1}{X}] \leq \frac{\xi}{K-1}$. Hence we have

$$\begin{aligned}
\left\|\boldsymbol{r}_n^P\big|_{D_n^c}\right\|_1 \leq & \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} K \sum_{l\geq l_n} (K-1)^l \left\|\mathbf{K}_n\right\|_\infty^l \\
\leq & \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} K \sum_{l\geq l_n} (K-1)^l \frac{\xi^l}{2^l(K-1)^l} \left\|\mathbf{P}\right\|_\infty \\
\leq & \sum_{|\mathbf{x}-\mathbf{x}_j|\leq d_n} \boldsymbol{k}_{nj} K \sum_{l\geq l_n} \frac{\xi^l}{2^l} o(1) \\
\leq & (\frac{\xi}{2})^{l_n} o(1) \\
= & o(\frac{1}{n})
\end{aligned}$$

□

F Proof for Theorem 2

Combining all discussions above, we present the main result Theorem 2 of this paper.

Proof. Define $c^D = \frac{\lambda}{1+\lambda q}$ and $c^P = 1$. To begin with, we will show that for $\forall \mathbf{x} \in [0,1]^d$,

$$\frac{\langle \boldsymbol{r}_n^{D,P}, f(\mathbf{X}_n)\rangle - c^{D,P}\boldsymbol{f}(\mathbf{x})}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \xrightarrow{\mathbb{P}} 0.$$

Let $\Delta^{D,P} = c^{D,P} - \sum_{i=1}^n \boldsymbol{r}_{n,i}^{D,P} = O\left(n^{-1}\right)$ and $\widetilde{\boldsymbol{f}}(\mathbf{x}) = (\boldsymbol{f}(\mathbf{x}), \ldots, \boldsymbol{f}(\mathbf{x}))^\top$ an n-vector. We consider the following decomposition:

$$\begin{aligned}
\frac{\langle \boldsymbol{r}_n^{D,P}, f(\mathbf{X}_n)\rangle - c^{D,P}\boldsymbol{f}(\mathbf{x})}{\left\|\boldsymbol{r}_n^{D,P}\right\|} &= \frac{\langle \boldsymbol{r}_n^{D,P}, \boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|} - \frac{\Delta^{D,P}\cdot \boldsymbol{f}(\mathbf{x})}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \\
&= -\frac{\Delta\cdot \boldsymbol{f}(\mathbf{x})}{\left\|\boldsymbol{r}_n^{D,P}\right\|} + \frac{\langle \boldsymbol{r}_n^{D,P}\big|_{D_n}, [\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n}\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|} + \frac{\langle \boldsymbol{r}_n^{D,P}\big|_{D_n^c}, [\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n^c}\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|}.
\end{aligned}$$

By 4, we have

$$\|\boldsymbol{k}_n\| = \Theta_p(n^{-\frac{1}{d+1}}), \left\|\boldsymbol{r}_n^{D,P}\right\| = \Theta_p(n^{-\frac{1}{2}\frac{1}{d+1}}).$$

Notice that

$$\left|\langle \boldsymbol{r}_n^{D,P}\big|_{D_n^c}, [\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n^c}\rangle\right| \leq \left\|\boldsymbol{r}_n^{D,P}\big|_{D_n^c}\right\|_1 \cdot \left\|[\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n^c}\right\|_\infty = O_p\left(\frac{1}{n}\cdot 2M_{\boldsymbol{f}}\right) = O_p\left(n^{-1}\right).$$

Hence we have the second term

$$\frac{\langle \boldsymbol{r}_n^{D,P}\big|_{D_n}, [\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n}\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \xrightarrow{\mathbb{P}} 0.$$

And by an argument earlier $|\Delta^{D,P}| = O\left(n^{-1}\right)$, thus

$$\frac{\Delta\cdot \boldsymbol{f}(\mathbf{x})}{\|\boldsymbol{r}_n\|} \xrightarrow{\mathbb{P}} 0.$$

Meanwhile, we can show similarly $|D_n| = \Omega_p\left(n\cdot(l_n\cdot d_n)^d\right)$ a.s.. And recall our underlying function is α-Lipischitz, therefore

$$\frac{\langle \boldsymbol{r}_n^{D,P}\big|_{D_n^c}, [\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n^c}\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \leq \frac{\left\|\boldsymbol{r}_n^{D,P}\big|_{D_n}\right\| \cdot \left\|[\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n}\right\|}{\left\|\boldsymbol{r}_n^{D,P}\right\|}$$

$$
\begin{aligned}
&\leq \left\| [\boldsymbol{f}(\mathbf{X}_n) - \widetilde{\boldsymbol{f}}(\mathbf{x})]\big|_{D_n} \right\| \\
&= O_p\left(\sqrt{n \cdot (l_n d_n)^d \cdot (l_n d_n \cdot \alpha)^2}\right) \\
&= O_p\left(\sqrt{n \cdot (\log n)^{d+2} \cdot d_n^{d+2}}\right) \\
&= O_p\left(\sqrt{n \cdot (\log n)^{d+2} \cdot n^{-\frac{d+2}{d+1}}}\right) \\
&= O_p\left((\log n)^{\frac{d+2}{2}} n^{-\frac{1}{2}\frac{1}{d+1}}\right).
\end{aligned}
$$

Therefore

$$
\frac{\langle \boldsymbol{r}_n^{D,P}, \boldsymbol{f}(\mathbf{X}_n)\rangle - c^{D,P}\boldsymbol{f}(\mathbf{x})}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \xrightarrow{\mathbb{P}} 0.
$$

Slutsky's Theorem then yields

$$
\frac{\widehat{\boldsymbol{f}}_n^{D,P}(\mathbf{x}) - c^{D,P}\boldsymbol{f}(\mathbf{x})}{\left\|\boldsymbol{r}_n^{D,P}\right\|} = \frac{\widehat{\boldsymbol{f}}_n(\mathbf{x}) - \langle \boldsymbol{r}_n^{D,P}, \boldsymbol{f}(\mathbf{X}_n)\rangle}{\left\|\boldsymbol{r}_n^{D,P}\right\|} + \frac{\langle \boldsymbol{r}_n^{D,P}, \boldsymbol{f}(\mathbf{X}_n)\rangle - c^{D,P}\boldsymbol{f}(\mathbf{x})}{\left\|\boldsymbol{r}_n^{D,P}\right\|} \xrightarrow{d} \mathcal{N}(0, \sigma_\epsilon^2).
$$

□

G Coverage Rates and Risk Bounds Guarantees

In this section we provide the proofs for coverage rates and risk bounds guarantees we gave in 5.

G.1 Proof for Corollary 1

Proof. The corollary holds by noticing that the variance of the scaled prediction of Algorithm 1 is $(\frac{1+\lambda q}{\lambda})^2 \left\|\boldsymbol{r}_n^D\right\|^2 \sigma^2$. Plugging in the rate of $\left\|\boldsymbol{r}_n^D\right\|$ by lemma 1 yields the risk bound. Similarly this can be shown for Algorithm 2. □

G.2 Proof for Corollary 3

Proof. We will show this convergence for the confidence interval Algorithm 1. For Algorithm 2, replacing the asymptotic variance and the final rescale constant will follow the same proof.

Denote the staged empirical coverage rate of the built-in confidence interval at any point $\mathbf{x} \in [0,1]^d$ as

$$
1 - \hat{\alpha}_{n,b}^{\text{CI}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\Big[\boldsymbol{f}(\mathbf{x}) \in \big[(c^D \widehat{\boldsymbol{f}}_b(\mathbf{x}) - c^D \left\|\boldsymbol{r}_n^D\right\| z_{1-\alpha/2}\sigma, c^D \widehat{\boldsymbol{f}}_b(\mathbf{x}) + c^D \left\|\boldsymbol{r}_n^D\right\| z_{1-\alpha/2}\sigma\big]\Big]
$$

Notice that with $b \to \infty$ we have $\widehat{\boldsymbol{f}}_b \overset{a.s.}{\to} \widehat{f}_\infty$. Hence, sending b to infinity and we drop the subscript for b to denote the coverage at infinite epoch.

$$
1 - \hat{\alpha}_{n}^{\text{CI}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\Big[\boldsymbol{f}(\mathbf{x}) \in \big[(c^D \widehat{\boldsymbol{f}}_\infty(\mathbf{x}) - c^D \left\|\boldsymbol{r}_n^D\right\| z_{1-\alpha/2}\sigma, c^D \widehat{\boldsymbol{f}}_\infty(\mathbf{x}) + c^D \left\|\boldsymbol{r}_n^D\right\| z_{1-\alpha/2}\sigma\big]\Big]
$$

which is equivalent to

$$
1 - \hat{\alpha}_{n}^{\text{CI}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\Big[\frac{\widehat{\boldsymbol{f}}_\infty(\mathbf{x}) - c^D\boldsymbol{f}(\mathbf{x})}{\|r_n^D\|\,\sigma} \in \big[-z_{1-\alpha/2}, z_{1-\alpha/2}\big]\Big]
$$

Since the indicator function is always finite, by SLLN:

$$
1 - \hat{\alpha}_{n}^{\text{CI}}(\mathbf{x}) \overset{a.s.}{\to} \mathbb{E}\left[\mathbb{1}\Big[\frac{\widehat{\boldsymbol{f}}_\infty(\mathbf{x}) - c^D\boldsymbol{f}(\mathbf{x})}{\|r_n^D\|\,\sigma} \in \big[-z_{1-\alpha/2}, z_{1-\alpha/2}\big]\Big]\right]
$$

$$= \mathbb{P}\Big[\frac{\widehat{\boldsymbol{f}}_\infty(\mathbf{x}) - c^D \boldsymbol{f}(\mathbf{x})}{\|r_n^D\|\,\sigma} \in \big[-z_{1-\alpha/2}, z_{1-\alpha/2}\big]\Big]$$
$$= 1-\alpha$$

For prediction intervals, we observe data according to the model

$$\mathbf{y} \;=\; \boldsymbol{f}(\mathbf{x}) \;+\; \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0,\sigma^2),$$

and our infinite-epoch estimator satisfies the CLT

$$\widehat{\boldsymbol{f}}_\infty(\mathbf{x}) \sim \mathcal{N}\big(c^D\,\boldsymbol{f}(\mathbf{x}),\, \|\boldsymbol{r}_n^D\|^2\,\sigma^2\big) \quad \text{as} \quad b\to\infty.$$

Since $\widehat{f}_\infty(\mathbf{x})$ and the new noise ε are independent Gaussians, their sum

$$\mathbf{y} = \boldsymbol{f}(\mathbf{x}) - c^D\widehat{\boldsymbol{f}}_\infty(\mathbf{x}) + c^D\widehat{\boldsymbol{f}}_\infty(\mathbf{x}) + \epsilon$$
$$= \mathcal{N}(c^D\widehat{\boldsymbol{f}}_\infty(\mathbf{x}), (c^D\,\|\boldsymbol{r}_n^D\|)^2\sigma^2 + \sigma^2)$$

And we produce prediction interval with the empirical coverage rate evaluated at:

$$1-\hat{\alpha}^{\text{PI}}_{n,b}(\mathbf{x})$$
$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\Big[\boldsymbol{f}(\mathbf{x}) \in \big[(c^D\widehat{\boldsymbol{f}}_b(\mathbf{x}) - \sqrt{(c^D\,\|\boldsymbol{r}_n^D\|)^2+1}\cdot z_{1-\alpha/2}\sigma, c^D\widehat{\boldsymbol{f}}_b(\mathbf{x}) + \sqrt{(c^D\,\|\boldsymbol{r}_n^D\|)^2+1}\cdot z_{1-\alpha/2}\sigma\big]\Big]$$

By similar argument above, sending $n, b \to \infty$ gives us almost sure convergence.

Regarding the reproduction interval, since we will have two independent model following the CLT in Theorem 2, their difference will also follow a CLT centered at zero with a variance inflated by a factor of 2 by independence. So we construct the reproduction interval as such:

$$1-\hat{\alpha}^{\text{RI}}_{n,b}(\mathbf{x})$$
$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\Big[\widehat{\boldsymbol{f}}_b^{(2)}(\mathbf{x}) \in \big[\widehat{\boldsymbol{f}}_b^{(1)}(\mathbf{x}) - \sqrt{2}(c^D\,\|\boldsymbol{r}_n^D\|)z_{1-\alpha/2}\sigma, \widehat{\boldsymbol{f}}_b^{(1)}(\mathbf{x}) + \sqrt{2}(c^D\,\|\boldsymbol{r}_n^D\|)z_{1-\alpha/2}\sigma\big]\Big]$$

And by a similar argument we can prove almost sure convergence. □

G.3 Proof for Corollary 2

Proof. To give a PAC argument, we decompose our error into statistical error $\widehat{\boldsymbol{f}}_*(\mathbf{x}) - \boldsymbol{f}(\mathbf{x})$ and algorithmic error $\widehat{\boldsymbol{f}}_b(\mathbf{x}) - \widehat{\boldsymbol{f}}_*(\mathbf{x})$.

From the Theorem 2, $\Pr\big(|\widehat{f}_*(\mathbf{x}) - f(\mathbf{x})| > \varepsilon\big) \le 2\exp\big(-\frac{\varepsilon^2}{2(c^{D,P})^2}\,n^{1/(d+1)}\big)$. Under the stated lower bound on n this probability is $\le \delta/2$.

To quantify the algorithmic error, we wish to make use of the Komolgorov inequality in Theorem 3.

For $\forall \mathbf{x} \in [0,1]^d$, let $\widehat{\boldsymbol{f}}_b(\mathbf{x})$ be the stage-b predictor, let $\widehat{\boldsymbol{f}}_*(\mathbf{x})$ be the fixed point defined in Theorem 1, and write

$$\mathbf{z}_b(\mathbf{x}) = \widehat{\boldsymbol{f}}_b(\mathbf{x}) - \widehat{\boldsymbol{f}}_*(\mathbf{x}), \qquad \epsilon_b = \mathbf{z}_b - \mathbb{E}[\mathbf{z}_b \mid \mathcal{F}_{b-1}],$$

It holds that the process $\mathbf{z}_b(\mathbf{x})$ is eventually also a stochastic contraction mapping by the exact same argument we made in proving Theorem 1, only replacing $\mathbf{K}_n$ with $\boldsymbol{k}_n(\mathbf{x})$ which still have row sum smaller than 1. And even we only have in probability guarantee of the staged mean contraction condition hold, since we are doing truncations and we can get rid off it after sufficiently large b, we can still check out the mean contraction condition $\sum_{b=1}^{\infty}(1-\lambda_b) = \infty$. Hence we invoke the Kolmogorov Inequality given in 3.

Consider the decomposition:

$$\Pr\big(\sup_{t\ge b}|\mathbf{z}_t| > \varepsilon\big) \le \Pr\big(|\mathbf{z}_b| > \tfrac{\varepsilon}{2}\big) + \Pr\big(\sup_{t\ge b}|\mathbf{z}_t| > |\mathbf{z}_b| + \tfrac{\varepsilon}{2}\big)$$

Because the SCM conditions are satisfied from b onward and $\sum_{t>b}\mathbb{E}[\epsilon_t^2] \le C^2/b$, , where $C = 2\lambda M(1+\sqrt{n})$. Theorem 3 gives the Kolmogorov's inequality below. Specifically, we can

choose b such that the numerator $\min\{\frac{\epsilon}{2}, \beta_{\frac{\epsilon}{2}}\} \geq \frac{\epsilon}{2}$ for suffieciently b', where $\beta_{\frac{\epsilon}{2}} = \frac{\epsilon}{2} + \|\mathbf{z}_t\| - \sqrt{d}\sup_{t>s}\|\epsilon_s\|$, when $b = \Omega\left(\frac{n\lambda^2 m^2}{\epsilon^2 \delta}\right)$. Hence

$$\Pr\big(\sup_{t\geq b}|\mathbf{z}_t| > |\mathbf{z}_b| + \tfrac{\varepsilon}{2}\big) \leq \frac{4\cdot 1\cdot (C^2/b)}{\min\{\frac{\epsilon}{2}, \beta_{\frac{\epsilon}{2}}\}^2} \leq \frac{4\cdot 1\cdot (C^2/b)}{(\varepsilon/2)^2} \;=\; \frac{16C^2}{\varepsilon^2 b}.$$

Since $|\epsilon_k| \leq C$, $V_b = \sum_{k=1}^{b} \mathbb{E}[\epsilon_k^2] \leq C^2\pi^2/6 < 2C^2$.

Set the martingale sum $S_b = \sum_{k=1}^{b}\epsilon_k$. Because $|\mathbf{z}_t| \leq |\mathbf{z}_{t-1}| + |\epsilon_t|$, telescoping gives

$$|\mathbf{z}_b| \;\leq\; |\mathbf{z}_0| + |S_b| \;\leq\; M + |S_b|.$$

Thus the event $\{|\mathbf{z}_b| > \varepsilon/2\}$ implies $|S_b| > (\varepsilon - 2M)/2$.

Freedman's maximal inequality for increments bounded by C and quadratic variation $\leq 2C^2$ yields

$$\Pr\Big(|\mathbf{z}_b| > \tfrac{\varepsilon}{2}\Big) \;\leq\; \Pr\Big(|S_b| > \tfrac{\varepsilon - 2M}{2}\Big) \;\leq\; 2\exp\Big(-\frac{(\varepsilon - 2M)_+^2}{16\,C^2}\Big).$$

Then the non-asymptotic algorithmic error bound:

$$\Pr\Big(|\mathbf{z}_b(\mathbf{x})| > \varepsilon\Big) \;\leq\; 2\exp\Big(-\frac{(\varepsilon - 4M)_+^2}{16\,C^2}\Big) \;+\; \frac{16\,C^2}{\varepsilon^2 b}, \quad C = 2\lambda M(1+\sqrt{n}).$$

With the stated lower bound on b this is $\leq \delta/2$. □

H Miscellanious Lemmas

Theorem 5 (Kolmogorov's Extension Theorem)**.** *Define the random variables:*

- $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ...) \in [0,1]^{d\times\mathbb{N}}$
- $\vec{\epsilon} = (\epsilon_1, \epsilon_2, \cdots) \in \mathbb{R}^{\mathbb{N}}$

The probability measure on $[0,1]^{d\times\mathbb{N}}$ ***and*** $\mathbb{R}^{\mathbb{N}}$ ***are uniquely determined by their product measures on the cylinder spaces****, reflecting the i.i.d. sampling of* $\mathbf{X}$ *and* $\vec{\epsilon}$*, such that*

$$\mathbf{y}_i = \boldsymbol{f}(\mathbf{x}_i) + \epsilon_i, \quad i \in \mathbb{N}$$

Lemma 8. *Let $X : \Omega_1 \to S$ and $\epsilon : \Omega_2 \to S$ be independent random variables. Assume $\{f_n : S \times S \to \mathbb{R}\}$ is a sequence of measurable functions. Suppose that for almost surely $x \in \Omega_1$ and $\epsilon \in \Omega_2$,*

$$f_n(x, \epsilon) \xrightarrow{d} N(0,1).$$

Then:

$$f_n(X, \epsilon) \xrightarrow{d} N(0,1).$$

Or equivalently

$$\lim_{n\to\infty} \mathbb{P}(f_n(X, \vec{\epsilon})) = \Phi(t)$$

Proof.

$$\lim_n \mathbb{P}(f_n(X,\epsilon) \leq t) = \lim_n \int\int \mathbb{1}_{\{f_n(x,\epsilon)\leq t\}}\, d\mu_x\, d\mu_\epsilon \tag{5}$$

$$= \lim_n \int \mathbb{P}(f_n(x,\epsilon) \leq t)\, d\mu_x \tag{6}$$

$$= \int \lim_n \mathbb{P}(f_n(x,\epsilon) \leq t)\, d\mu_x \tag{7}$$

$$= \int \Phi(t)\, d\mu_x = \Phi(t). \tag{8}$$

(8) is justified by the Fubini theorem since the indicator function is always non-negative. And (9) is guaranteed by Dominated Convergence Theorem since the integrand is always bounded (with assumption of it being a sub-Gaussian density of ϵ). □

I Mean Squared Error on UCI Machine Learning Repository

We provide further results for the MSE of our methods, benchmarks, and competitors on six additional UCI datasets. The dataset we are using in this work are from: Nash et al. (1994), Vito (2008), Schlimmer (1985), est (2019), Cortez (2008), Cortez et al. (2009), Redmond (2002).

The hyperparameters chosen by Optuna are listed below:

Table 2: Abalone

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.0213	0.0249	0.0108	—	0.5476	0.5750	0.2724	—
max_depth	3	3	9	11	10	16	3	—
subsample	—	0.6332	—	—	—	—	—	—
num_leaves	—	—	37	—	—	—	—	—
alpha	—	—	—	—	—	—	—	0.1299
l1_ratio	—	—	—	—	—	—	—	0.3753
min_samples_split	—	—	—	—	43	42	5	—
subsample_rate	—	—	—	—	0.5071	0.5133	0.6266	—
dropout_rate	—	—	—	—	0.4766	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	16	—

Table 3: Automobile

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.2019	0.2153	0.2341	—	0.9411	0.1416	0.5671	—
max_depth	3	3	8	14	6	11	3	—
subsample	—	0.9850	—	—	—	—	—	—
num_leaves	—	—	37	—	—	—	—	—
alpha	—	—	—	—	—	—	—	0.3634
l1_ratio	—	—	—	—	—	—	—	0.1386
min_samples_split	—	—	—	—	23	9	30	—
subsample_rate	—	—	—	—	0.9368	0.9899	0.7344	—
dropout_rate	—	—	—	—	0.2145	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	16	—

Table 4: Communities and Crime

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.0224	0.0379	0.0215	—	0.6904	0.5772	0.3578	—
max_depth	6	3	5	16	15	16	4	—
subsample	—	0.5375	—	—	—	—	—	—
num_leaves	—	—	20	—	—	—	—	—
alpha	—	—	—	—	—	—	—	3.8079
l1_ratio	—	—	—	—	—	—	—	0.9556
min_samples_split	—	—	—	—	35	49	5	—
subsample_rate	—	—	—	—	0.7385	0.7209	0.5148	—
dropout_rate	—	—	—	—	0.1903	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	11	—

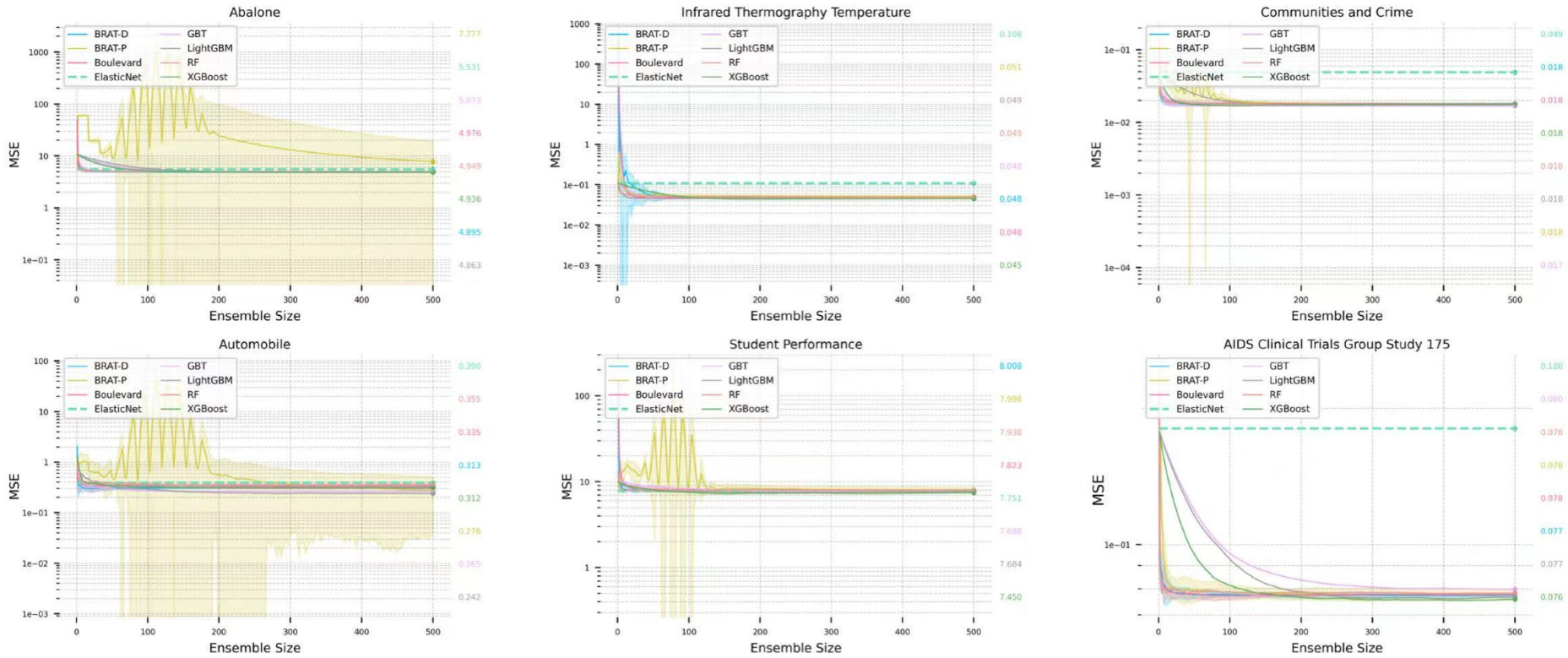

Figure 6: Mean Squared Error on UCI Machine Learning Repositories

Table 5: Wine Quality

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.0731	0.1139	0.1186	—	0.8396	0.8905	0.8200	—
max_depth	10	10	16	20	15	15	16	—
subsample	—	0.6980	—	—	—	—	—	—
num_leaves	—	—	42	—	—	—	—	—
alpha	—	—	—	—	—	—	—	3.8079
l1_ratio	—	—	—	—	—	—	—	0.9556
min_samples_split	—	—	—	—	4	2	13	—
subsample_rate	—	—	—	—	0.7317	0.7431	0.6669	—
dropout_rate	—	—	—	—	0.2762	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	10	—

Table 6: Student Performance

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.0177	0.1836	0.0218	—	0.9256	0.8698	0.9824	—
max_depth	3	5	5	11	3	9	3	—
subsample	—	0.5780	—	—	—	—	—	—
num_leaves	—	—	38	—	—	—	—	—
alpha	—	—	—	—	—	—	—	0.4855
l1_ratio	—	—	—	—	—	—	—	0.1373
min_samples_split	—	—	—	—	39	32	48	—
subsample_rate	—	—	—	—	0.8418	0.5249	0.5964	—
dropout_rate	—	—	—	—	0.1230	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	13	—

Table 7: Obesity Level

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.1905	0.0916	0.0834	—	0.2487	0.5382	0.3643	—
max_depth	7	7	16	16	16	16	11	—
subsample	—	0.8121	—	—	—	—	—	—
num_leaves	—	—	25	—	—	—	—	—
alpha	—	—	—	—	—	—	—	0.2182
l1_ratio	—	—	—	—	—	—	—	0.1200
min_samples_split	—	—	—	—	3	3	4	—
subsample_rate	—	—	—	—	0.6466	0.7153	0.5595	—
dropout_rate	—	—	—	—	0.8079	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	4	—

Table 8: AIDS Clinical Trials Group Study 175

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.0177	0.0226	0.0108	—	0.8519	0.9137	0.3010	—
max_depth	3	3	9	5	4	4	4	—
subsample	—	0.7530	—	—	—	—	—	—
num_leaves	—	—	37	—	—	—	—	—
alpha	—	—	—	—	—	—	—	3.8079
l1_ratio	—	—	—	—	—	—	—	0.9556
min_samples_split	—	—	—	—	12	27	46	—
subsample_rate	—	—	—	—	0.5335	0.7374	0.5342	—
dropout_rate	—	—	—	—	0.3421	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	3	—

Table 9: Infrared Thermography Temperature

Hyperparameter	GBT	XGBoost	LightGBM	RF	BRAT-D	Boulevard	BRAT-P	ElasticNet
n_estimators	500	500	500	500	500	500	500	—
learning_rate	0.0289	0.0167	0.1836	—	0.7616	0.6558	0.9864	—
max_depth	5	3	1	20	3	3	3	—
subsample	—	0.6988	—	—	—	—	—	—
num_leaves	—	—	24	—	—	—	—	—
alpha	—	—	—	—	—	—	—	3.8079
l1_ratio	—	—	—	—	—	—	—	0.9556
min_samples_split	—	—	—	—	4	4	9	—
subsample_rate	—	—	—	—	0.5055	0.5438	0.6371	—
dropout_rate	—	—	—	—	0.2395	0.0000	—	—
n_trees_per_group	—	—	—	—	—	—	2	—

The learning rate for BRAT-P is only valid in the first vanilla boosting round. After that, all trees are fixed with learning rate 1.

J Raincloud for Interval Coverage

Following the discussion in Section 6, we present more raincloud plots below:

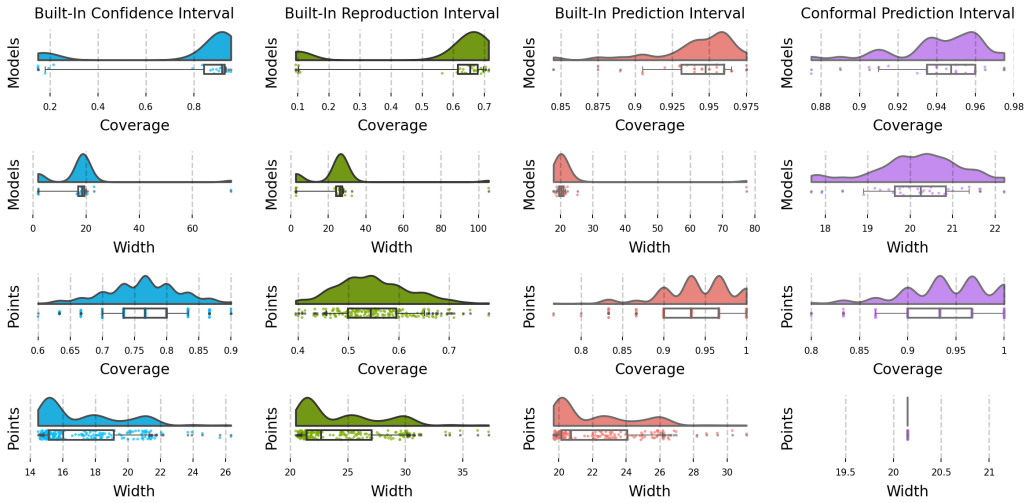

Figure 7: Moderate coverage, where there is neither overfitting nor underfitting.

(a) Test size: 200; Model Replications: 30; Ensemble Size: 200; Learning Rate: 0.6; Subsampling Rate: 0.8; Dropout Rate: 0.3; Maximum Depth: 4; Nystrom subsampling rate: 0.1.

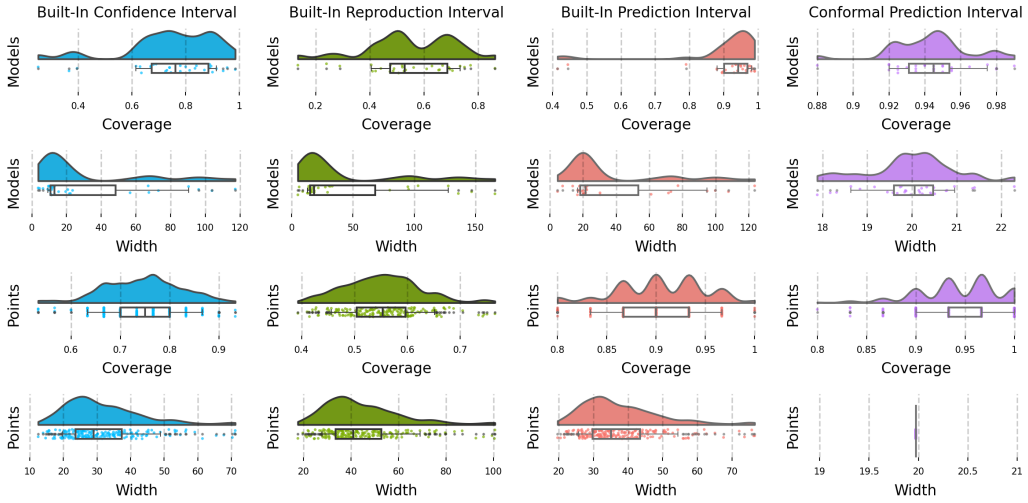

Figure 8: Wide coverage, when there is overfitting.

(a) Test size: 200; Model Replications: 30; Ensemble Size: 200; Learning Rate: 0.3; Subsampling Rate: 0.8; Dropout Rate: 0.3; Maximum Depth: 6; Nystrom subsampling rate: 0.1.

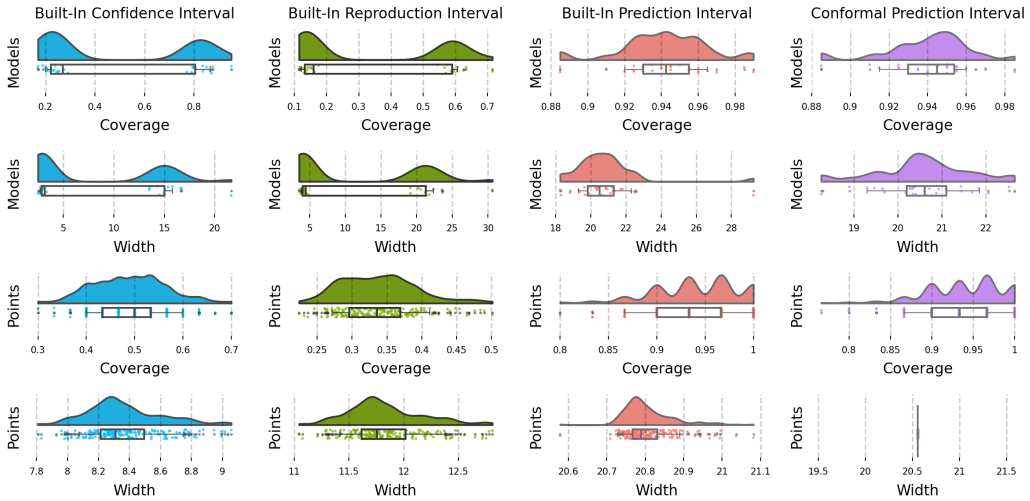

Figure 9: Narrow coverage, when there is underfitting.

(a) Test size: 200; Model Replications: 30; Ensemble Size: 200; Learning Rate: 1.0; Subsampling Rate: 0.8; Dropout Rate: 0.9; Maximum Depth: 4; Nystrom subsampling rate: 0.1.

K Visualizations of Interval Coverage on a 1D Signal

As an extension of Section 6, we display more visualizations of our intervals on a 1D interval.

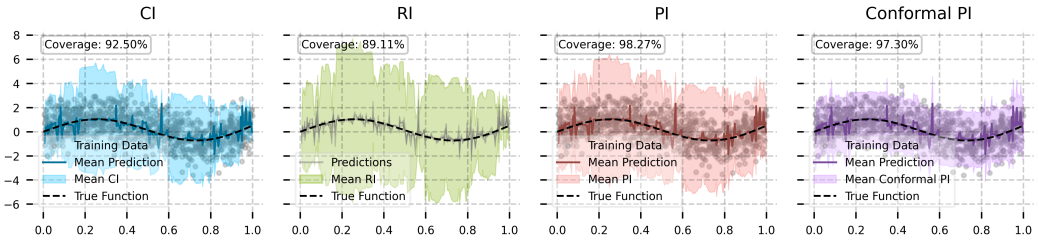

Figure 10: Moderate coverage, neither overfitting nor underfitting.

(a) Ensemble Size: 300; Learning Rate: 1.0; Maximum Depth: 8; Subsampling Rate: 0.9; Dropout rate: 0.6

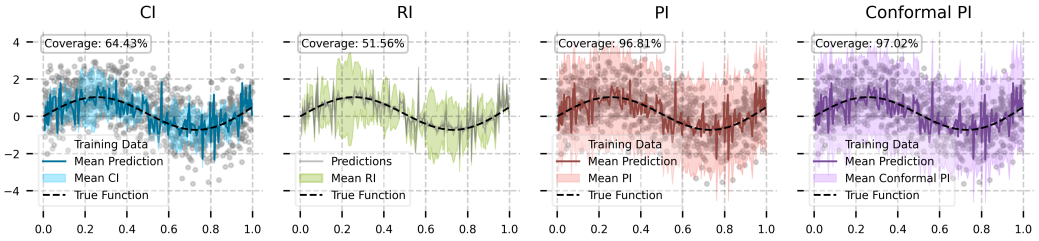

Figure 11: Low coverage, overfitting.

(a) Ensemble Size: 100; Learning Rate: 0.3; Maximum Depth: 12; Subsampling Rate: 0.6; Dropout rate: 0.3

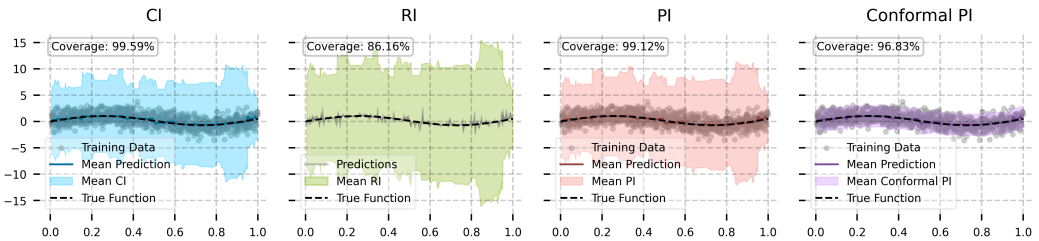

Figure 12: High coverage, underfitting.

(a) Ensemble Size: 100; Learning Rate: 0.3; Maximum Depth: 4; Subsampling Rate: 0.6; Dropout rate: 0.6

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: The abstract and introduction state that we (1) propose both random-dropout and structured-dropout boosting algorithms, (2) derive their asymptotic normality via a CLT, and (3) establish finite-sample statistical rates. These claims correspond directly to algorithms definitions and CLT statement in Section 3 (proof in Appendix E). Statistical rates are given in Section 4 (proofs in Appendix G).
   Guidelines:
   - The answer NA 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 NA 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 explicitly acknowledge in Section 7 that our theoretical guarantees rest on strong assumptions (structure–value isolation, non-adaptivity, and tree regularity) which may not hold in practice, and we outline the need to relax these; we also note that our methods are currently limited to regression and discuss challenges in extending to classification, survival analysis, and structured outputs.
   Guidelines:
   - The answer NA 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: The paper mainly introduced two modified boosting algorithms: Algorithm 1 and Algorithm 2. In Section 2 and 5 we introduced all assumptions needed for our following results to hold. Both algorithms are shown to converge to a fixed point in Appendix C. And the corresponding derivations of CLTs are provided in Appendix E.
Guidelines:

- The answer NA 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: All parameters used to produce the presented results are either provided in Section 6, or provided in the appendix.
   Guidelines:
   - The answer NA 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: We include all data-processing scripts, model training and evaluation code, and a README with exact commands and environment specifications in the supplementary materials. All UCI datasets used are public and instructions for download and preprocessing are provided.
Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?
Answer: [Yes]
Justification: We describe our train/test splits, evaluation protocol, and Optuna-based hyperparameter optimization in Section 6, with full model settings tabulated in Appendix I.
Guidelines:
   - The answer NA 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: In MSE races, we run models for 5 replications and fill the mean MSE trajectories $+/- 2\widehat{s.d.}$. In coverage rate simulations we provide kernel density estimations.
Guidelines:
   - The answer NA 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: All experiments were run on an Apple MacBook Pro (2022, Apple M2, 8 GB RAM); individual runs took under 2 hours and the full suite under 6 hours. These modest requirements mean the results can be reproduced on any modern laptop.

   Guidelines:

   - The answer NA 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: All authors have reviewed the NeurIPS Code of Ethics and will preserve anonymity.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: This work develops statistical inference tools for gradient boosting in tabular regression—a foundational methodological advance without a specific application domain or direct deployment context. As such, there is no clear pathway to societal harms or benefits tied exclusively to our contributions.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work focuses on statistical inference for gradient boosting on tabular regression tasks, involving no pretrained networks, scraped media, or high-risk generative models, and thus does not pose significant misuse risks.

    Guidelines:

    - The answer NA 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: We cite each UCI dataset by name and reference (all are publicly available under the UCI Machine Learning Repository terms), and we reference the original papers for all libraries and methods used (e.g., XGBoost, LightGBM).

    Guidelines:

    - The answer NA 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: [NA]

Justification: We do not introduce or release any new datasets, codebases, or pre-trained models—our experiments use publicly available UCI datasets and standard libraries.

Guidelines:

- The answer NA 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: [NA]

Justification: All experiments are done on sythetic dataset or existing dataset from UCI repository.

Guidelines:

- The answer NA 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: [NA]

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

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: All core methodological components were designed and implemented by the authors. Any use of LLMs was limited to writing, editing, or formatting and did not influence the scientific content or originality of the work.

Guidelines:

- The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.