Local Linear Forests

Rina Friedberg
rsf@stanford.edu

Julie Tibshirani
julietibs@gmail.com

Susan Athey
athey@stanford.edu

Stefan Wager
swager@stanford.edu

November 9, 2018

Abstract

Random forests are a powerful method for non-parametric regression, but are limited in their ability to fit smooth signals, and can show poor predictive performance in the presence of strong, smooth effects. Taking the perspective of random forests as an adaptive kernel method, we pair the forest kernel with a local linear regression adjustment to better capture smoothness. The resulting procedure, *local linear forests*, enables us to improve on asymptotic rates of convergence for random forests with smooth signals, and provides substantial gains in accuracy on both real and simulated data. We prove a central limit theorem and propose a computationally efficient construction for confidence intervals.

1 Introduction

Random forests [Breiman, 2001] are a popular method for non-parametric regression that have proven effective across many application areas [Cutler, Edwards Jr, Beard, Cutler, Hess, Gibson, and Lawler, 2007, Díaz-Uriarte and De Andres, 2006, Svetnik, Liaw, Tong, Culberson, Sheridan, and Feuston, 2003]. A major weakness of random forests, however, is their inability to exploit smoothness in the regression surface they are estimating. As an example, consider the following setup with a smooth trend in the conditional response surface: We simulate $x_1, \ldots, x_n$ independently from the uniform distribution on $[0, 1]^{20}$, with responses

$$y = \log\left(1 + \exp(6x_1)\right) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,\, 20), \tag{1}$$

and our goal is to estimate $\mu(x) = \mathbb{E}[Y \,|\, x = x]$. The left panel of Figure 1 shows a set of predictions on this data from a random forest. The forest is unable to exploit strong local trends and, as a result, fits the target function using qualitatively the wrong shape: The prediction surface resembles a step function as opposed to a smooth curve.

In order to address this weakness, we take the perspective of random forests as an adaptive kernel method. This interpretation follows work by Athey, Tibshirani, and

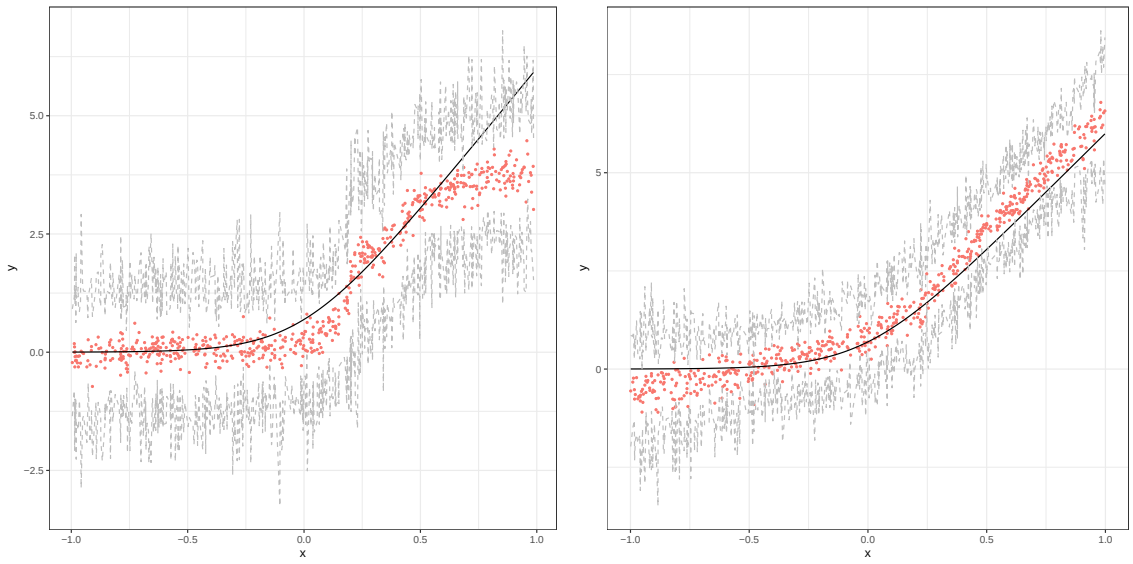

Figure 1: Example 95% confidence intervals from generalized random forests (left) and local linear forests (right) on out of bag predictions from equation 1. Training data were simulated from equation (1), with $n = 500$ training points, dimension $d = 20$ and errors $\varepsilon \sim N(0, 20)$. Forests were trained using the `R` *package* `grf` *[Tibshirani, Athey, Wager, Wright, and all contributors to the included version of Eigen, 2018] and tuned via cross-validation. True signal is shown as a smooth curve, with dots corresponding to forest predictions, and upper and lower bounds of pointwise confidence intervals connected in the dashed lines. Here the data was generated with $n = 500, \sigma = \sqrt{20}$, and $d = 20$, with subsampling rate $s/n = 0.5$.*

Wager [2018], Hothorn, Lausen, Benner, and Radespiel-Troger [2004], and Meinshausen [2006], and complements the traditional view of forests as an ensemble method (i.e., an average of predictions made by individual trees). Specifically, random forest predictions can be written as

$$\hat{\mu}_{\text{rf}}(x) = \sum_{i=1}^{n} \alpha_i(x)\, Y_i, \quad \sum_{i=1}^{n} \alpha_i(x) = 1, \quad \alpha_i(x) \geq 0, \tag{2}$$

where the weights $\alpha_i(x)$ defined in (6) encode the weight given by the forest to the i-th training example when predicting at x. Now, as is well-known in the literature on non-parametric regression, if we want to fit smooth signals without some form of neighborhood averaging (e.g., kernel regression, k-NN, or matching for causal inference), it is helpful to use a local regression adjustment to correct for potential misalignment between a test point and its neighborhood [Abadie and Imbens, 2011, Cleveland and Devlin, 1988, Fan and Gijbels, 1996, Heckman, Ichimura, and Todd, 1998, Loader, 1999, Newey, 1994, Stone, 1977, Tibshirani and Hastie, 1987]. These types of adjustments are particularly important near boundaries, where neighborhoods are asymmetric by necessity, but with many covariates, the adjustments are also important away from boundaries given that local neighborhoods are often unbalanced due to sampling variation.

The goal of this paper is to improve the accuracy of forests on smooth signals using regression adjustments, potentially in many dimensions. By using the local regression

adjustment, it is possible to adjust for asymmetries and imbalances in the set of nearby points used for prediction, ensuring that the weighted average of the feature vector of neighboring points is approximately equal to the target feature vector, and that predictions are centered. The use of regression adjustments further implies that the remaining prediction error is due to strong local curvature in the data generating process. The improvement to forests from the regression adjustment is most likely to be large in cases where some features have strong effects with moderate curvature, so that regression adjustments are both effective and important. Many datasets have this characteristic; for example, labor market outcomes tend to improve with parents' educational and labor market attainment, but there are diminishing returns.

In their simplest form, local linear forests just take the forest weights $\alpha_i(x)$, and use them for local regression:

$$\begin{pmatrix} \hat{\mu}(x) \\ \hat{\theta}(x) \end{pmatrix} = \text{argmin}_{\mu,\theta} \left\{ \sum_{i=1}^{n} \alpha_i(x)(Y_i - \mu(x) - (x_i - x)\theta(x))^2 + \lambda||\theta(x)||_2^2 \right\}. \quad (3)$$

Here $\hat{\mu}(x)$ is estimates the conditional mean function $\mu(x)$, and $\theta(x)$ corrects for the local trend in $x_i - x$. The ridge penalty $\lambda||\theta(x)||_2^2$ prevents overfitting to the local trend, and plays a key role both in simulation experiments and asymptotic convergence results. Then, as discussed in Section 2.1, we can improve the performance of local linear forests by modifying the tree-splitting procedure used to get the weights $\alpha_i(x)$, and making it account for the fact that we will use local regression to estimate $\mu(x)$. In our software, we choose all tuning parameters, including minimum node size and λ, via cross-validation. As a first encouraging result we note that, in the motivating example from Figure 1, local linear forests have essentially eliminated the bias of standard forests.

From a formal perspective, we study local linear forests using asymptotic theory. Theorem 1 gives a Central Limit Theorem for the predictions $\hat{\mu}(x)$ from a local linear forest at a given test point x, specifying the asymptotic convergence rate and its dependence on subsampling and smoothness of $\mu(x)$. This result allows us to build pointwise Gaussian confidence intervals using the delta method, giving practitioners applicable uncertainty quantification for local linear forest predictions. Observe that in Figure 1, the bias from regression forest predictions affects not only the prediction curve but also the corresponding confidence intervals, which are not centered on the true function. Local linear forests, in addressing this bias issue, improve over regression forests in both predictive performance and confidence interval coverage. Strikingly, our local linear forest confidence intervals simultaneously achieve better coverage and are shorter than confidence intervals built using regression forests.

A simple form of (3), without regularization or modified tree-splitting procedures, was also considered in a recent paper by Bloniarz, Talwalkar, Yu, and Wu [2016]. However, Bloniarz, Talwalkar, Yu, and Wu [2016] only report modest performance improvements over basic regression forests; for example, on the "Friedman function" they report roughly a 5% reduction in mean-squared error. In contrast, we find fairly large, systematic improvements from local linear forests; see, e.g., Figure 5 for corresponding results on the same Friedman function. It thus appears that our algorithmic modifications via regularization and optimized splitting play a qualitatively important role in getting local linear forests to work well. These empirical findings are also mirrored in our theory. For example, in order to prove rates of convergence for local linear forests

that can exploit smoothness of $\mu(\cdot)$ and improve over corresponding rates available for regression forests, we need an appropriate amount of regularization in (3).

Finally, we note that one can also motivate local linear forests from the starting point of local linear regression. Despite working well in low dimensions, classical approaches to local linear regression are not applicable to even moderately high-dimensional problems.[1] In contrast, random forests are adept at fitting high-dimensional signals, both in terms of their stability and computational efficiency. From this perspective, random forests can be seen as an effective way of producing weights to use in local linear regression. In other words, local linear forests aim to combine the strength of random forests in fitting high dimensional signals, and the ability of local linear regression to capture smoothness. We explore the properties of this procedure from both a theoretical and an empirical perspective, and find that it presents a promising alternative to traditional random forests.

1.1 Motivating Example: Attitudes to Welfare

For conciseness, this paper focuses on local linear forests for non-parameteric regression; however, as discussed in Section 2.4, a similar local linear correction can also be applied to quantile regression forests [Meinshausen, 2006], causal forests [Wager and Athey, 2018] or, more broadly, to any instance of generalized random forests [Athey, Tibshirani, and Wager, 2018]. To highlight this potential, we start our discussion with an example of heterogeneous treatment effect estimation using local linear causal forests.

As in Wager and Athey [2018], we frame our discussion in terms of the Neyman-Rubin causal model [Imbens and Rubin, 2015]. Suppose we have data (x_i, Y_i, W_i), where x_i are covariates, $Y_i \in \mathbb{R}$ is the response, and $W_i \in \{0, 1\}$ is the treatment. In order to define the causal effect of the treatment W_i, we posit potential outcomes for individual i, $Y_i(0)$ and $Y_i(1)$, corresponding to the response the subject would have experienced in the control and treated conditions respectively; we then observe $Y_i = Y_i(W_i)$. We seek to estimate the conditional average treatment effect (CATE) of W, namely $\tau(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]$. Assuming uncounfoundedness [Rosenbaum and Rubin, 1983],

$$\{Y_i(0), Y_i(1)\} \perp\!\!\!\perp W_i \,|\, x_i, \tag{4}$$

the CATE is in general identified via non-parametric methods. We assume unconfoundedness when discussing treatment effect estimation through this paper. Wager and Athey [2018] proposed an extension of random forests for estimating CATEs, and Athey, Tibshirani, and Wager [2018] improved on the method by making it locally robust to confounding using the transformation of Robinson [1988]. Here, we propose a local linear correction to the method of Athey, Tibshirani, and Wager [2018] to strengthen its performance when $\tau(\cdot)$ is smooth; see Section 2.4 for details.

To illustrate the value of the local linear causal forests, we consider a popular dataset from the General Social Survey (GSS) that explores how word choice reveals public opinions about welfare [**?**]. Individuals filling out the survey from 1986 to 2010 answered whether they believe the government spends too much, too little, or the right amount

[1] The curse of dimensionality for kernel weighting is well known. The popular core `R` function `loess` [R Core Team, 2018] allows only 1-4 predictors, while `locfit` [Loader, 2013] crashes on the simulation from (1) with $d \geq 7$.

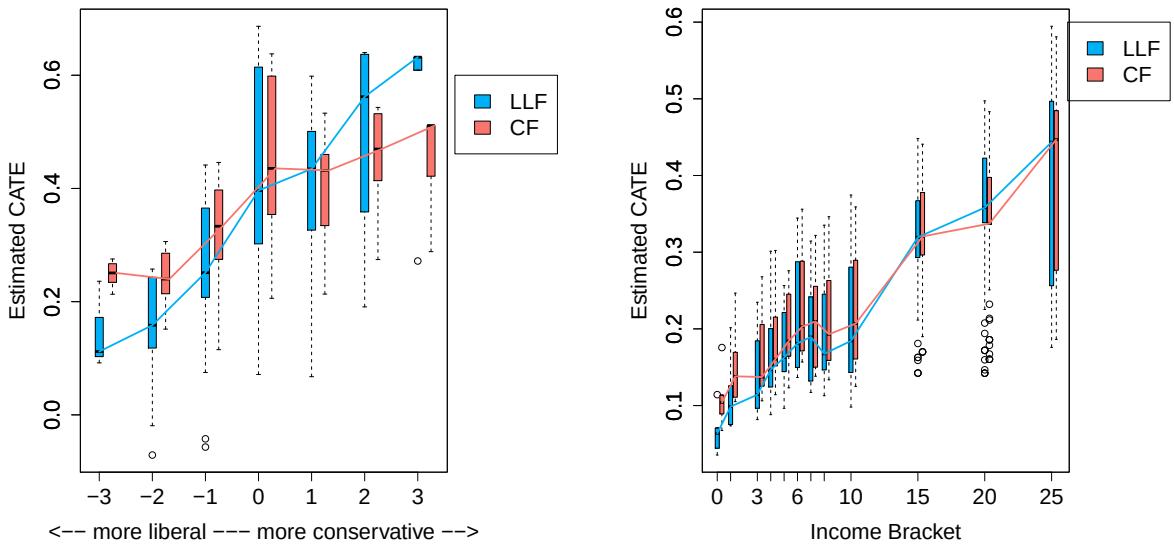

Figure 2: Estimated trends in CATE of welfare language by political views (left) and by income (right). On the x-axis is the bracket of either political views or income, and the boxplots show test set predictions $\hat{\tau}(x_i)$ from local linear causal forests and causal forests. Lines connect the medians of each boxplot, dashed for causal forests and solid for local linear causal forests. Each forest was trained on 400 subsampled training points, with cross-validation, and evaluated on 1000 test points.

on the social safety net. [2] GSS randomly assigned the wording of this question, such that the social safety net was either described as "welfare" or "assistance to the poor". This change had a well-documented effect on responses due to the negative perception many Americans have about welfare; moreover, there is evidence of heterogeneity in the CATE surface [Green and Kern, 2012].

Here, we write $W_i = 1$ if the i-th sample received the "welfare" treatment, and define $Y_i = 1$ if the i-th response was that the government spends too much on the social safety net. Thus, a positive treatment effect $\tau(x)$ indicates that, conditionally on $x_i = x$, using the phrase "welfare" as opposed to "assistance to the poor" increases the likelihood that the i-th subject says the government spends too much on the social safety net. We base our analysis on $d = 12$ covariates, including income, political views, age, and number of children. The full dataset has $N = 28,646$ observations; here, to make the problem interesting, we test our method on smaller subsamples of the data.

Figure 2 displays the CATE by categories of income and political views, as estimated by both a local linear causal forest and a plain causal forest using $n = 600$ observations, again both implemented using `grf`. The display suggests that local linear causal forests are able to exploit smoothness and monotonicity to reveal more heterogeneity along these important features, which may mean that the local linear correction is helping.

In order to compare the performance of both methods more formally, we use the

[2]The authors acknowledge that NORC at the University of Chicago, and all parties associated with the General Social Survey (including the principal investigators), offer the data and documentation "as is" with no warranty and assume no legal liability or responsibility for the completeness, accuracy, or usefulness of the data, or fitness for a particular purpose.

Subsample size	200	400	800	1200	1500	2000
CF	0.035	0.021	0.015	0.014	0.011	0.007
LLCF	0.027	0.017	0.013	0.013	0.011	0.006

Table 1: Estimated MSE of estimating the treatment effect on subsampled welfare data, averaged over 200 runs at each subsample size. Here we have calculated the estimated in sample MSE (5). We show estimated test error from local linear causal forests (LLCF) and standard causal forests (CF). Tuning parameters were selected via cross-validation using the R-learner objective.

transformed outcome metric of Athey and Imbens [2016]. Noting that $\mathbb{E}[(2W_i - 1)Y_i] = \tau(x_i)$, they suggest examining the following test set error criterion

$$\begin{aligned} \mathcal{E} &= \frac{1}{|\mathcal{S}_{test}|} \sum_{i \in \mathcal{S}_{test}} \left((2W_i - 1)Y_i - \hat{\tau}(x_i)\right)^2, \\ \mathbb{E}[\mathcal{E}] &= \mathbb{E}\left[\left(\tau(X) - \hat{\tau}(X)\right)^2\right] + S_0, \quad S_0 = \mathbb{E}\left[\left((2W_i - 1)Y_i - \tau(x_i)\right)^2\right]. \end{aligned} \tag{5}$$

If we can estimate S_0, then (5) gives an unbiased estimate of the mean-squared error of $\hat{\tau}(\cdot)$. Here, we estimate S_0 via out-of-bag estimation on the full dataset with $N = 28,646$, assuming that a local linear forest with such a large sample size has negligible error.

Table 1 has error estimates for both types of forests using (5), and verifies that using the local linear correction improves empirical performance across different subsample sizes. Section 4 contains a more detailed simulation study of local linear causal forests, comparing them with a wider array of baseline methods.

1.2 Related Work

Random forests were first introduced by Breiman [2001], building on the work of Breiman, Friedman, Stone, and Olshen [1984] on recursive partitioning (CART), Breiman [1996] on bagging, and Amit and Geman [1997] on randomized trees. Bühlmann and Yu [2002] shows how the bagging makes forests smoother than single trees, while Biau [2012] and Scornet, Biau, and Vert [2015] establishes asymptotic risk consistency of random forests under specific assumptions. More sophisticated tree-based ensembles motivated by random forests have been proposed by Basu, Kumbier, Brown, and Yu [2018], who iteratively grow feature-weighted tree ensembles that perform especially well for discovering interactions, Zhou and Hooker [2018], who consider a hybrid between random forests and boosting, and Zhu, Zeng, and Kosorok [2015], who do deeper search during splitting to mitigate the greediness of CART.

The idea of considering random forests as an adaptive kernel method has been proposed by several papers. Hothorn, Lausen, Benner, and Radespiel-Troger [2004] suggest using weights from survival trees and gives compelling simulation results, albeit to our knowledge no theoretical guarantees. Meinshausen [2006] proposes this technique for quantile regression forests and gives asymptotic consistency of the resulting predictions. Athey, Tibshirani, and Wager [2018] leverage this idea to present generalized random forests as a method for solving heterogeneous estimating equations.

They derive an asymptotic distribution and confidence intervals for the resulting predictions. Local linear forests build on this literature; the difference being being that we use the kernel-based perspective on forests to exploit smoothness of $\mu(\cdot)$ rather than to target more complicated estimands (such as a quantile).

Early versions of confidence intervals for random forests, backed by heuristic arguments and empirical evidence, were proposed by Sexton and Laake [2009] and Wager, Hastie, and Efron [2014]. Mentch and Hooker [2016] then established asymptotic normality of random forests where each tree depends on a small subsample of training examples, while Wager and Athey [2018] provide an asymptotic characterization of forests that allows for asymptotically valid confidence intervals. The confidence intervals proposed here are motivated by the algorithm of Sexton and Laake [2009], paired with the random forest delta method developed by Athey, Tibshirani, and Wager [2018] for statistical inference with generalized random forests.

As mentioned in the introduction, a predecessor to this work is a paper by Bloniarz, Talwalkar, Yu, and Wu [2016], who consider local linear regression with supervised weighting functions, including ones produced by a forest. The main differences between our method and that of Bloniarz, Talwalkar, Yu, and Wu [2016] is that they do not adapt the tree-splitting procedure to account for the local linear correction, and do not consider algorithmic features—such as ridge penalization—that appear to be needed to achieve good performance both in theory and in practice. Additionally, our method is flexible to forests targeting any heterogeneous estimating equation, and in particular to causal forests. On the formal side, Bloniarz, Talwalkar, Yu, and Wu [2016] prove consistency of their method; however, they do not establish rates of convergence and thus, unlike in our Theorem 1, they cannot use smoothness of $\mu(\cdot)$ to improve convergence properties of the forest. They also do not provide a central limit theorem or confidence intervals.

More broadly, there is an extensive body of work on model-based trees that explores different combinations of local regression and trees. Torgo [1997] and Gama [2004] study functional models for tree leaves, fitting models instead of local averages at each node. Karalič [1992] suggests fitting a local linear regression in each leaf, and Torgo [1997] highlights the performance of kernel methods in general for MOB tree methods. Zeileis, Hothorn, and Hornik [2008], and later Rusch and Zeileis [2013], propose not only prediction, but recursive partitioning via fitting a separate model in each leaf, similar to the residual splitting strategy of local linear forests. Local linear forests complement this literature; they differ, however, in treating forests as a kernel method. The leaf nodes in a local linear forest serve to provide neighbor information, and not local predictions.

Our work is motivated by the literature on local linear regression and maximum likelihood estimation [Abadie and Imbens, 2011, Cleveland and Devlin, 1988, Fan and Gijbels, 1996, Heckman, Ichimura, and Todd, 1998, Loader, 1999, Newey, 1994, Stone, 1977, Tibshirani and Hastie, 1987]. Stone [1977] introduces local linear regression and gives asymptotic consistency properties. Cleveland [1979] expands on this by introducing robust locally weighted regression, and Fan and Gijbels [1992] give a variable bandwidth version. Cleveland and Devlin [1988] explore further uses of locally weighted regression. Local linear regression has been particularly well-studied for longitudinal data, as in Li and Hsing [2010] and Yao, Muller, and Wang [2005]. Cheng, Fan, and Marron [1997] use local polynomials to estimate the value of a function at the bound-

ary of its domain. Abadie and Imbens [2011] show how incorporating a local linear correction improves nearest neighbor matching procedures.

2 Local Linear Forests

Local linear forests use a random forest to generate weights that can then serve as a kernel for local linear regression. Suppose we have training data $(x_1, Y_1), \ldots, (x_n, Y_n)$ with $Y_i = \mu(x_i) + \varepsilon_i$, for $\varepsilon_i \sim N(0, \sigma^2)$. Consider using a random forest to estimate the conditional mean function $\mu(x) = \mathbb{E}[Y \mid X = x]$ at a fixed test point x_0. Traditionally, random forests are viewed as an ensemble method, where tree predictions are averaged to obtain the final estimate. Specifically, for each tree T_b in a forest of B trees, we find the leaf $L_b(x_0)$ with predicted response $\hat{\mu}_b(x_0)$, which is simply the average response of all training data points assigned to $L_b(x_0)$. We then predict the average $\hat{\mu}(x_0) = (1/B) \sum_{b=1}^{B} \hat{\mu}_b(x_0)$.

An alternate angle, advocated by Athey, Tibshirani, and Wager [2018], Hothorn, Lausen, Benner, and Radespiel-Troger [2004], and Meinshausen [2006], entails viewing random forests as adaptive weight generators, as follows. Equivalently write $\hat{\mu}_b(x_0)$ as

$$\begin{aligned}\hat{\mu}(x_0) &= \frac{1}{B} \sum_{b=1}^{B} \sum_{i=1}^{n} Y_i \frac{1\{x_i \in L_b(x_0)\}}{|L_b(x_0)|} \\ &= \sum_{i=1}^{n} Y_i \frac{1}{B} \sum_{b=1}^{B} \frac{1\{x_i \in L_b(x_0)\}}{|L_b(x_0)|} = \sum_{i=1}^{n} \alpha_i(x_0) Y_i,\end{aligned}$$

where the forest weight $\alpha_i(x_0)$ is

$$\alpha_i(x_0) = \frac{1}{B} \sum_{b=1}^{B} \frac{1\{x_i \in L_b(x_0)\}}{|L_b(x_0)|} \tag{6}$$

Notice that by construction, $\sum_{i=1}^{n} \alpha_i(x_0) = 1$ and for each i, $0 \leq \alpha_i(x_0) \leq 1$. Athey, Tibshirani, and Wager [2018] use this perspective to harness random forests for solving weighted estimating equations, and give asymptotic guarantees on the resulting predictions.

Local linear forests solve the locally weighted least squares problem (3) with weights (6). Note that equation (3) has a closed-form solution, as follows. Throughout this paper, we let A be the diagonal matrix with $A_{i,i} = \alpha_i(x_0)$, and let J denote the $d+1 \times d+1$ diagonal matrix with $J_{1,1} = 0$ and $J_{i+1,i+1} = A_{i,i}$, so as to not penalize the intercept. We define Δ, the centered regression matrix with intercept, as $\Delta_{i,1} = 1$ and $\Delta_{i,j+1} = x_{i,j} - x_{0,j}$. Then the local linear forest estimator can be explicitly written as

$$\begin{pmatrix} \hat{\mu}(x_0) \\ \hat{\theta}(x_0) \end{pmatrix} = \left(\Delta^T A \Delta + \lambda J\right)^{-1} \Delta^T A Y \tag{7}$$

Qualitatively, we can think of local linear regression as a weighting estimator with a modulated weighting function $\gamma_i \alpha_i(x_0)$ whose x-moments are better aligned with the test point x_0: $\hat{\mu}(x_0) = \sum_{i=1}^{n} \gamma_i \alpha_i(x_0) Y_i$ with $\sum_{i=1}^{n} \gamma_i \alpha_i(x_0) = 1$ and $\sum_{i=1}^{n} \gamma_i \alpha_i(x_0) x_i \approx x_0$, where the last relation would be exact without a ridge penalty (i.e., with $\lambda = 0$).

With the perspective of generating a kernel for local linear regression in mind, we move to discuss the appropriate splitting rule for local linear forests.

2.1 Splitting for Local Regression

Random forests traditionally use Classification and Regression Trees (CART) from Breiman, Friedman, Stone, and Olshen [1984] splits, which proceed as follows. We consider a parent node P with n_P observations $(x_{i1}, Y_{i1}), \ldots, (x_{in_P}, Y_{in_P})$. For each candidate pair of child nodes C_1, C_2, we take the mean value of Y inside each child, $\bar{Y}_1$ and $\bar{Y}_2$. Then we choose C_1, C_2 to minimize the sum of squared errors

$$\sum_{i:x_i \in C_1} (Y_i - \bar{Y}_i)^2 + \sum_{i:x_i \in C_2} (Y_i - \bar{Y}_2)^2$$

Knowing that we will use the forest weights to perform a local regression, we neither need nor want to use the forest to model strong, smooth signals; the final regression step can model the strong global effects. Instead, in the parent node P, we run a ridge regression to predict Y_{ik} from X_{ik}.

$$\hat{Y}_{ik} = x_{ik}^T \hat{\beta}_P, \quad \text{for } \hat{\beta}_P = (x_P^T x_P + \lambda J)^{-1} x_P^T Y_P \tag{8}$$

We then run a standard CART split on the residuals $Y_{ik} - \hat{Y}_{ik}$, modeling local effects in the forest and regressing global effects back in at prediction. Observe that, much like the CART splitting rule, an appropriate software package can enforce that a forest using this splitting rule splits on every variable and gives balanced splits; hence this splitting rule may be used to grow honest and regular trees.

To explore the effects of CART and residual splitting rules, we consider this simulation first introduced by Friedman [1991]. Generate $x_1, \ldots, x_n$ independently and identically distributed $U[0,1]^5$ and model Y_i from

$$y = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10 x_4 + 5 x_5 + \epsilon, \tag{9}$$

for $\varepsilon \sim N(0, \sigma^2)$. This model has become a popular study for evaluating nonparametric regression methods; see for example Chipman, George, and McCulloch [2010] and Taddy, Chen, Yu, and Wyle [2015]. It is a natural setup to test how well an algorithm handles interactions $\sin(\pi x_1 x_2)$, its ability to pick up a quadratic signal $20(x_3 - 0.5)^2$, and how it simultaneously models strong linear signals $10x_4 + 5x_5$.

Figure 3 displays the split frequencies from an honest random forest (left) using standard CART splits, and a local linear forest (right). The x-axis is indexed by variable, here x_1 through x_5, and the y-axis gives tree depth for the first 4 levels of tree splits. Tiles are colored according to how often trees in the forest split on that variable; a darker tile denotes more splits at that tree depth. CART splits very frequently on x_4, which contributes the strongest linear signal, especially at the top of the tree but consistently throughout levels. Local linear forests, on the other hand, rarely split on either of the strong linear signals, instead spending splits on the three that are more difficult to model. In Section 4, we show that this indeed corresponds to a performance gain from local linear forests.

2.2 Honest Forests

Unless noted otherwise, all random forests used in this paper are grown using a type of sub-sample splitting called "honesty", used by Wager and Athey [2018] to derive the asymptotic distribution of random forest prediction. As outlined in Procedure

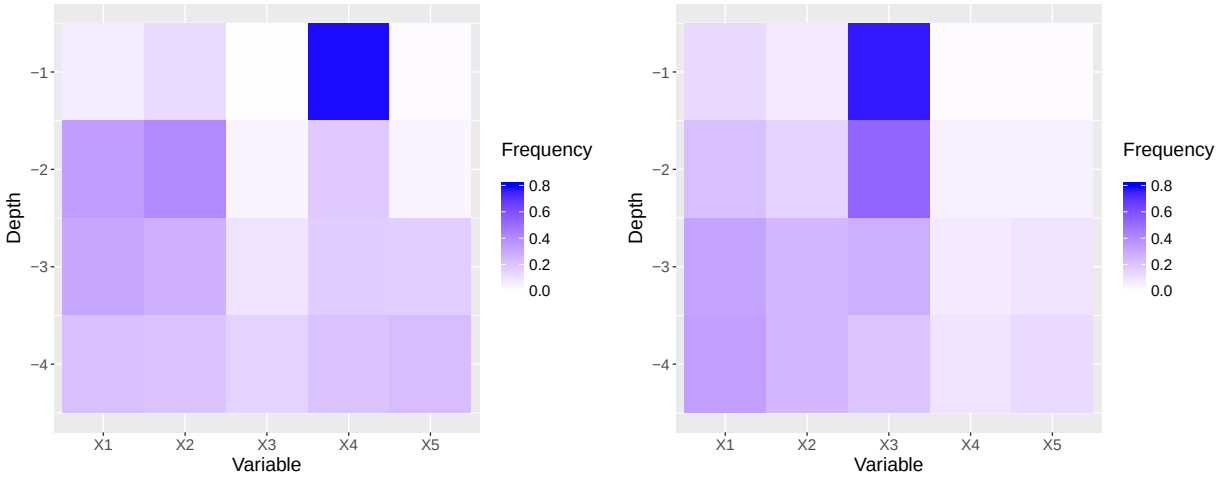

Figure 3: Split frequency plot for CART splits from an honest random forest (left) and residual splits from a local linear forest (right). Each forest was trained on $n = 600$ observations from the data-generating process in 9. Variables 1 through 5 are on the x-axis, and the y-axis gives tree depth, starting with depth 1 at the top of the plot. Tile color is according to split frequency, so variables on which the forest splits frequently at depth j have a dark tile in row j.

1 of Wager and Athey [2018], each tree in an honest forest is grown using two non-overlapping subsamples of the training data, call them $\mathcal{I}_b$ and $\mathcal{J}_b$. We first choose a tree structure T_b using only the data in $\mathcal{J}_b$, and write $x \leftrightarrow_b x'$ as the boolean indicator for whether the points x and x' fall into the same leaf of T_b. Then, in a second step, we define the set of neighbors of x as $L_b(x) = \{i \in \mathcal{I}_b : x \leftrightarrow_b x_i\}$; this neighborhood function is what we then use to define the forest weights in (6).

This type of subsample-splitting lets us control for potential overfitting when growing the tree T_b, because the samples $\mathcal{J}_b$ used to define the neighborhood $L_b(x)$ were held out when growing T_b. We note that, despite considerable interest in the literature, there are no available consistency results for random forests with fully grown trees that do not use honesty. Biau [2012] uses a different type of sample splitting, Biau, Devroye, and Lugosi [2008] and Wager and Walther [2015] rely on large leaves, while the results of Scornet, Biau, and Vert [2015] on fully grown trees rely on an unchecked high-level assumption. Thus, we choose to build our forests with honesty on by default.

Perhaps the strongest criticism of honesty is that, because of the additional subsample splitting step, we end up with trees grown on less data that are thus less expressive. However, if this is a concern, it is possible to use only a small fraction of the data for the set $\mathcal{I}_b$, since these samples are used to generate weights and not directly for estimation. By growing a sufficiently large number of trees, we can ensure that the weights are stable. To the extent that concerns about sample size remain, local linear forests appear to be a particularly natural companion to honesty, as we can use the final local regression step to compensate for potential undersmoothing of the raw forests.

2.3 Tuning a Local Linear Forest

We recommend selecting ridge penalties by cross-validation. It is often reasonable to choose different values of λ for forest training and for local linear prediction. During forest growth, equation (8) gives ridge regression coefficients $\hat{\beta}_P$ in each parent leaf. As trees are grown on subsamples of data, over-regularization at this step is a danger even in large leaves. Consequently, small values of λ are advisable for penalization on regressions during forest training. Furthermore, as we move to small leaves, computing meaningful regression coefficients becomes more difficult; the ridge regression can begin to mask signal instead of uncovering it. A heuristic that performs well in practice is to store the regression estimates $\hat{\beta}_P$ on parent leaves P. When the child leaf size shrinks below a cutoff, we use $\hat{\beta}_P$ from the parent node to calculate ridge residual pseudo-outcomes, instead of estimating them from unstable regression coefficients on the small child dataset. In practice, this helps to avoid the pitfalls of over-regularizing and of regressing on a very small dataset when growing the forest. At the final regression prediction step (7), however, a larger ridge penalty can control the variance and better accommodate noisy data.

With increasingly high-dimensional data, feature selection before prediction can significantly reduce error and decrease computation time. Often, a dataset will contain only a small number of features with strong global signals. In other cases, a researcher will known in advance which variables are consistently predictive or of special interest. In these cases, it is reasonable to run the regression prediction step on this smaller subset of predictors expected to contribute overarching trends. Such covariates, if they are not already known, can be chosen by a stepwise regression or lasso, or any other technique for automatic feature selection. Last, it is worth noting that these tuning suggestions are pragmatic in nature; the theoretical guarantees provided in Section 3 are for local linear forests trained without these heuristics.

2.4 Extension to Causal Forests

Finally, although we have focused on local linear regression forests, similar ideas also apply to other types of forests, such as quantile regression forests [Meinshausen, 2006] or, more broadly, generalized random forests [Athey, Tibshirani, and Wager, 2018]. Here, we discuss the case of the orthogonalized causal forests proposed by Athey, Tibshirani, and Wager [2018]; other cases are analogous.

Using notation from Section 1.1, we assume observations $(x_i,\, W_i,\, Y_i)$ where W_i is an unconfounded treatment assignment. Orthogonalized causal forests start by estimating the nuisance components

$$e(x) = \mathbb{P}[W_i = 1 \,|\, x_i = x] \text{ and } m(x) = \mathbb{E}[Y_i \,|\, x_i = x] \tag{10}$$

using a regression forest, and then estimate the CATE function $\tau(x)$ via a generalized random forest induced by the residual-on-residual estimating equation of Robinson [1988]; see also Nie and Wager [2017] for a broader discussion of this family of CATE estimators. Procedurally, given a forest kernel $\alpha_i(x)$, a causal forest then estimates the treatment effect as

$$\{\hat{\tau}(x),\, \hat{a}(x)\} = \text{argmin}_{\tau,\, a} \left\{ \sum_{i=1}^{n} \alpha_i(x) \left(Y_i - \hat{m}^{(-i)}(x_i) - a - \tau \left(W_i - \hat{e}^{(-i)}(x_i) \right) \right)^2 \right\}, \tag{11}$$

where the $^{(-i)}$-superscript denotes leave-one-out predictions from the nuisance models. If nuisance estimates are accurate, the intercept $\hat{a}$ should be 0; however, we leave it in the optimization for robustness.

Local linear causal forests are then a simple generalization of this idea. We first use local linear forests to fit the nuisance components (10), and then add a regularized adjustment to (11),

$$\left\{\hat{\tau}(x),\, \hat{\theta}_\tau(x),\, \hat{a}(x),\, \hat{\theta}_a(x)\right\} = \text{argmin}_{\tau,\,\theta} \left\{ \sum_{i=1}^{n} \alpha_i(x) \Big(Y_i - \hat{m}^{(-i)}(x_i) - a - (x_i - x)\theta_a \right.$$
$$\left. - \left(\tau + \theta_\tau (x_i - x)\right) \left(W_i - \hat{e}^{(-i)}(x_i)\right) \Big)^2 + \lambda_\tau \left\|\theta_\tau\right\|_2^2 + \lambda_a \left\|\theta_a\right\|_2^2 \right\}. \tag{12}$$

We cross-validate local linear causal forests (including λ_τ and λ_a) by minimizing the R-learning criterion recommended by Nie and Wager [2017]:

$$\widehat{\text{Err}}\left(\hat{\tau}(\cdot)\right) = \sum_{i=1}^{n} \left(Y_i - \hat{m}^{(-i)}(x_i) - \hat{\tau}(x_i) \left(W_i - \hat{e}^{(-i)}(x_i)\right)\right)^2. \tag{13}$$

3 Asymptotic Theory

Before we delve into the main result and its proof, we briefly discuss why the asymptotic behavior of locally linear forests cannot be directly derived from existing results on regression forests. This is due to a key difference in the dependence structure of the forest. In the regression case, a random forest prediction at x_0 is $\hat{\mu}_{\text{rf}}(x_0) = \sum_{i=1}^{n} \alpha_i(x_0) Y_i$, where, due to honesty, Y_i is independent of $\alpha_i(x_0)$ given x_i. This conditional independence plays a key role in the argument of Wager and Athey [2018]. Analogously to $\hat{\mu}_{\text{rf}}(x_0)$, we can write the local linear forest prediction as a weighted sum,

$$\hat{\mu}(x_0) = \sum_{i=1}^{n} \alpha_i(x_0) \rho_i, \quad \rho_i = e_1^T M_\lambda^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} Y_i, \quad M_\lambda = \Delta^T A \Delta + \lambda J, \tag{14}$$

where we use notation Δ, A, J from (7). At a first glance, $\hat{\mu}(x_0)$ indeed looks like the output of a regression forest trained on observations ρ_i. However, as highlighted in Figure 4, the dependence structure of this object is different. In a random forest, we make Y_i and $\alpha_i(x_0)$ independent by conditioning on x_i. For a local linear forest, however, conditioning on x_i will not guarantee that ρ_i and $\alpha_i(x_0)$ are independent, thus breaking a key component in the argument of Wager and Athey [2018].

We now give a Central Limit Theorem for local linear forest predictions, beginning by stating assumptions on the forest.

Assumption 1. (Regular Trees) We assume that the forest grows regular trees: that the trees are symmetric in permutations of training data index, split on every variable with probability bounded from below by $\pi > 0$, and are balanced in that each split puts at least a fraction $\omega > 0$ of parent observations into each child node.

Assumption 2. (Honest Forests) We assume that the forest is honest as described in Section 2.2, meaning that two distinct and independent samples are used to select the splits and estimate parameters in the nodes.

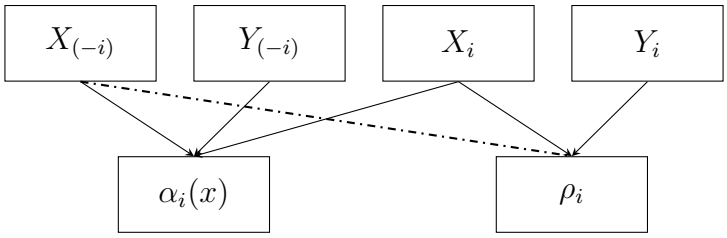

Figure 4: Visualization of the dependence structure in random forests as compared to local linear forests. Arrows indicate a dependence of the derived quantities, weights $\alpha_i(x)$ and pseudo-outcomes ρ_i, on data $X_{(-i)}, x_i, Y_{(-i)}$, and Y_i, where the $_{(-i)}$-subscript denotes all elements of a vector or matrix except for entry i. The solid lines represent dependencies in both random forests and local linear forests, and the dashed line is a dependence that only exists for local linear forests.

Subsampling plays a central role in our asymptotic theory, as this is what allows us to prove asymptotic normality by building on the work of Efron and Stein [1981]. Moreover, subsampling is what we use to tune the bias-variance trade-off of the forest: Forests whose trees are grown on small subsamples have higher bias but lower variance (and vice-versa).

In order to establish asymptotic unbiasedness of forests, Wager and Athey [2018] require a subsample size of at least $n^{\beta_{\text{rf}}}$, with

$$\beta_{\text{rf}} := 1 - \left(1 + \frac{d}{\pi} \frac{\log(\omega^{-1})}{\log((1-\omega)^{-1})}\right)^{-1} < \beta < 1. \tag{15}$$

This convergence rate of a traditional honest random forest, however, does not improve when $\mu(x)$ is smooth. Here, we show that by using a local regression adjustment and assuming smoothness of $\mu(\cdot)$, we can grow trees on smaller subsamples of size (16) without incurring asymptotic bias. This allows us to decrease the variance (and improve the accuracy) of our estimates.

Our main result establishes asymptotic normality of local linear forest predictions, and gives this improved subsampling rate. The rate given depends on our bounds on the squared radius of a leaf, detailed in Appendix B in the statement and proof of Lemma 6.

Theorem 1. *Suppose we have training data $Z_i = (x_i, Y_i)$ identically and independently distributed on $[0,1]^d \times \mathbb{R}$, where the density of x_i is bounded away from infinity. Suppose furthermore that $\mu(x) = \mathbb{E}[Y \mid x = x]$ is twice continuously differentiable, with a uniformly bounded second derivative, that $\text{Var}[Y \mid x = x] > 0$. Last, say that our trees are grown according to Assumptions 1 and 2, with subsamples of size s with $s = n^\beta$, for*

$$\beta_{\min} := 1 - \left(1 + \frac{d}{1.56\pi} \frac{\log(\omega^{-1})}{\log((1-\omega)^{-1})}\right)^{-1} < \beta < 1, \tag{16}$$

and that the ridge regularization parameter grows at rate

$$\lambda = \Theta\left(d\sqrt{\frac{s}{n}}\left(\frac{s}{2k-1}\right)^{-1.56\frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}}\right).$$

Then for each fixed test point x_0, there is a sequence $\sigma_n(x_0) \to 0$ such that

$$\frac{\hat{\mu}_n(x_0) - \mu(x_0)}{\sigma_n(x_0)} \Rightarrow N(0,1), \quad \sigma_n^2(x_0) = O(n^{-(1-\beta)})$$

3.1 Pointwise Confidence Intervals

Before giving a proof of Theorem 1, we discuss the derivation of our variance estimates. This section complements our main result, as the Central Limit Theorem becomes far more useful when we have valid standard error estimates. Following Athey, Tibshirani, and Wager [2018], we use the random forest delta method to develop pointwise confidence intervals for local linear forest predictions.

The random forest delta method starts from a solution to a local estimating equation with random forest weights $\alpha_i(x)$:

$$\sum_{i=1}^{n} \alpha_i(x) \psi\left(x_i,\, Y_i;\, \hat{\mu}(x),\, \hat{\theta}(x)\right) = 0. \tag{17}$$

Athey, Tibshirani, and Wager [2018] then propose estimating the error of these estimates as

$$\widehat{\mathrm{Var}}\left[\left(\hat{\mu}(x), \hat{\theta}(x)\right)\right] = \widehat{V}(x)^{-1} \widehat{H}_n(x) \left(\widehat{V}(x)^{-1}\right)', \tag{18}$$

where $V(x) = \partial/\partial(\mu,\theta)\mathbb{E}[\psi(x, Y; \mu, \theta \mid x = x)]$ is the slope of the expected estimating equation at the optimum, and

$$\widehat{H}_n(x) = \widehat{\mathrm{Var}}\left[\sum_{i=1}^{n} \alpha_i(x) \psi\left(x_i,\, Y_i;\, \mu^*(x),\, \theta^*(x)\right)\right]. \tag{19}$$

The upshot is that $\widehat{H}_n(x)$ measures the variance of an (infeasible) regression forest with response depending on the score function ψ at the optimal parameter values, and that we can in fact estimate $\widehat{H}_n(x)$ using tools originally developed for variance estimation with regression forests. Meanwhile, $V(x)$ can be estimated directly using standard methods.

With local linear forests, $(\hat{\mu}, \hat{\theta})$ solve (17) with score function

$$\psi(Y_i, x_i; \mu, \theta) = \nabla_{(\mu,\theta)} \left(\left(Y_i - \Delta_i \begin{pmatrix} \mu \\ \theta \end{pmatrix} \right)^2 + \lambda \left\| \theta \right\|_2^2 \right), \tag{20}$$

where we again use notation defined in (7) and (14). First, we note that we have access to a simple and explicit estimator for $V(x)$: Twice differentiating (3) with respect to the parameters $(\mu,\, \theta)$ gives

$$\nabla^2_{(\mu,\theta)} \left(\sum_{i=1}^{n} \alpha_i(x_0)(Y_i - \mu - \Delta_i\theta)^2 + \lambda ||\theta||_2^2 \right) = \sum_{i=1}^{n} \alpha_i(x_0) \Delta_i^T \Delta_i + \lambda J = M_\lambda, \tag{21}$$

which we can directly read off of the forest. In this paper, we are only interested in confidence intervals for $\mu(x)$, which can now be represented in terms of $\zeta' = e_1' M_\lambda^{-1}$:

$$\begin{aligned} \hat{\sigma}_n^2 &:= \zeta' \widehat{H}_n(x) \zeta = \widehat{\mathrm{Var}}\left[\sum_{i=1}^{n} \alpha_i(x)\; \Gamma_i(\mu^*(x),\, \theta^*(x))\right], \\ \Gamma_i(\mu,\, \theta) &= (\zeta \cdot \Delta_i) \left(Y_i - \Delta_i \begin{pmatrix} \mu \\ \theta \end{pmatrix} \right). \end{aligned} \tag{22}$$

Next, we follow Athey, Tibshirani, and Wager [2018], and proceed using the bootstrap of little bags construction of Sexton and Laake [2009]. At a high level this method is a computationally efficient half-sampling estimator. For any half sample $\mathcal{H}$, let $\Psi_{\mathcal{H}}$ be the average of the empirical scores Γ_i averaged over trees that only use data from the half-sample $\mathcal{H}$:

$$\Psi_{\mathcal{H}} = \frac{1}{|\mathcal{S}_{\mathcal{H}}|} \sum_{b \in \mathcal{S}_{\mathcal{H}}} \frac{\sum_{i=1}^{n} 1\left(\{x_i \in L_b(x)\}\right) \Gamma_i \left(\hat{\mu}(x),\, \hat{\theta}(x)\right)}{\sum_{i=1}^{n} 1\left(\{x_i \in L_b(x)\}\right)}, \tag{23}$$

where $\mathcal{S}_{\mathcal{H}}$ is the set of trees that only use data from the half-sample $\mathcal{H}$, and $L_b(x)$ contains neighbors of x in the b-th tree (throughout, we assume that the subsample used to grow each tree has less than $n/2$ samples). Then, a standard half-sampling estimator would simply use [Efron, 1982]

$$\hat{\sigma}_n^2 = \binom{n}{\lfloor n/2 \rfloor}^{-1} \sum_{\{\mathcal{H}\,:\,|\mathcal{H}|=\lfloor \frac{n}{2} \rfloor\}} \left(\Psi_{\mathcal{H}} - \bar{\Psi}\right)^2, \quad \bar{\Psi} = \binom{n}{\lfloor n/2 \rfloor}^{-1} \sum_{\{\mathcal{H}\,:\,|\mathcal{H}|=\lfloor \frac{n}{2} \rfloor\}} \Psi_{\mathcal{H}}. \tag{24}$$

Now, carrying out the full computation in (24) is impractical, and naive Monte Carlo approximations suffer from bias. However, as discussed in Athey, Tibshirani, and Wager [2018] and Sexton and Laake [2009], bias-corrected randomized algorithms are available and perform well. Here, we do not discuss these Monte Carlo bias corrections, and instead refer to Section 4.1 of Athey, Tibshirani, and Wager [2018] for details. Simulation results on empirical confidence interval performance are given in Section 4.3.

3.2 Supporting Results

We now give a proof of Theorem 1, beginning by decomposing $\hat{\mu}(x_0)$ into bias $\Delta_1(x_0)$ and variance $\hat{\gamma}_n(x_0)$, the latter of which we will approximate by a regression forest. Define the diameter (and corresponding radius) of a tree leaf as the length of the longest line segment that can fit completely inside of the leaf. Thanks to our assumed uniform bound on the second derivative of $\mu(\cdot)$, a Taylor expansion of $y = \mu(x) + \varepsilon$ around $\mu(x_0)$ yields the following decomposition,

$$\hat{\mu}(x) = \sum_{i=1}^{n} e_1^T M_\lambda^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \alpha_i(x_0) Y_i = \mu(x_0) + \Delta_1(x_0) + \hat{\gamma}_n(x_0) + O\left(\bar{R}^2\right), \tag{25}$$

where $\bar{R}^2$ is the average squared radius of leaves T_b in the forest, and we have isolated two error terms

$$\Delta_1(x) = \sum_{i=1}^{n} \alpha_i(x_0) e_1^T M_\lambda^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \nabla\mu(x_0)^T \begin{pmatrix} 0 \\ x_i - x_0 \end{pmatrix}, \tag{26}$$

$$\hat{\gamma}_n(x) = \sum_{i=1}^{n} \alpha_i(x_0) e_1^T M_\lambda^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \varepsilon_i. \tag{27}$$

For simplicity, moving forward we will write $\alpha_i(x_0) = \alpha_i$, dropping the written dependence on x_0.

To control the radius R_{T_b} of a typical leaf containing x_0 (and thus the Taylor error in (25)), we use the following bound. Suppose $x_1, \ldots, X_s \sim U([0,1]^d)$ independently, and let T_b be any regular, random-split tree and let $R_{T_b}(x_0)$ be the radius of its leaf containing the test point x_0. Consider Lemma 2 of Wager and Athey [2018], which states the following: Let T be a regular, random-split tree and let $L(x_0)$ denote its leaf containing x_0. Suppose that $x_1, \ldots, X_s \sim U([0,1]^d)$ independently. Then, for any $0 < \eta < 1$, and for large enough s,

$$\mathbb{P}\left[\text{diam}_j(L(x_0)) \geq \left(\frac{s}{2k-1}\right)^{-\frac{0.99(1-\eta)\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}}\right] \leq \left(\frac{s}{2k-1}\right)^{-\frac{\eta^2}{2}\frac{1}{\log(\omega^{-1})}\frac{\pi}{d}} \tag{28}$$

Choose $\eta = 1.25\sqrt{\log((1-\omega)^{-1})} \leq 0.6$, so that $1.98(1-\eta) \geq 0.79$, and $\eta^2/2 \geq 0.78\log((1-\omega)^{-1})$. Consequently, by setting $\eta = 1.25\sqrt{\log((1-\omega)^{-1})}$ in the above bound, for sufficiently large s we have

$$\mathbb{P}\left(R^2_{T_b}(x_0) \geq r_s\right) \leq d\left(\frac{s}{2k+1}\right)^{-0.78\frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}}, \tag{29}$$

$$\text{where } r_s = \sqrt{d}\left(\frac{s}{2k+1}\right)^{-0.79\frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}}.$$

Next, to control the behavior of $\hat{\gamma}_n(x_0)$, we show that M_λ concentrates around its expectation. The proof of Lemma 2 uses concentration bounds for U-statistics given by Hoeffding [1963].

Lemma 2. *Let $x_1, \ldots, x_n$ be independent and identically distributed on $[0,1]^d$. Let $\alpha_1, \ldots, \alpha_n$ be forest weights from trees grown on subsamples of size s and radius bounded by r_s from (29). Then,*

$$||M_\lambda - \mathbb{E}[M_\lambda]||_\infty = \mathcal{O}_P\left(r_s^2\sqrt{s/n}\right)$$

Lemma 2 enables coupling $\hat{\gamma}_n$ with an approximation $\tilde{\gamma}_n$, defined as

$$\tilde{\gamma}_n(x_0) = \sum_{i=1}^n \alpha_i \tilde{Y}_i, \quad \text{where} \quad \tilde{Y}_i = e_1^T \mathbb{E}[M_\lambda]^{-1}\begin{pmatrix}1\\ x_i - x\end{pmatrix}\varepsilon_i.$$

Now, $\tilde{Y}_i$ is independent of α_i conditionally on x_i (because the problematic associations in Figure 4 were mediated by M_λ), and consequently $\tilde{\gamma}_n$ can be characterized via standard tools used to study random forests.

Corollary 3. *Under the conditions from Lemma 2, $\hat{\gamma}_n(x_0)$ and $\tilde{\gamma}_n(x_0)$ are coupled at the following rate.*

$$|\hat{\gamma}_n(x_0) - \tilde{\gamma}_n(x_0)| = \mathcal{O}_P\left(\lambda^{-1} r_s^2\sqrt{s/n}\right)$$

Corollary 3 dictates our choice of λ at $\mathcal{O}(r_s^2\sqrt{s/n})$. For any $\lambda < \mathcal{O}(r_s^2\sqrt{s/n})$, $|\hat{\gamma}_n(x) - \tilde{\gamma}_n(x)|$ may not converge to 0. On the other hand, observe that letting $\lambda \to \infty$ will make local linear forest predictions equivalent to regression forest predictions. An appropriate choice of λ allows us to derive the improved asymptotic normality of $\tilde{\gamma}_n(x_0)$, given below in Lemma 4.

Lemma 4. *Suppose that trees T are honest, regular, and grown on subsamples of size s, with $s = n^\beta$ for some*

$$\beta > 1 - \left(1 + \frac{1}{1.56}\frac{\log(\omega^{-1})}{\log((1-\omega)^{-1})}\frac{d}{\pi}\right)^{-1} = \beta_{\min}$$

where π and ω are constants defined in the forest assumptions. Suppose further that observations $x_1, \ldots, x_n$ are i.i.d. on $[0,1]^d$ with a density f bounded away from infinity, and that the conditional mean function $\mu(x_0)$ is twice Lipschitz continuous at x_0. Last, suppose we choose $\lambda = \mathcal{O}(r_s^2\sqrt{s/n})$. Then there is a sequence $\sigma_n(x_0) \to 0$ such that

$$\frac{\tilde{\gamma}_n(x_0)}{\sigma_n(x_0)} \Rightarrow \mathcal{N}(0,1)$$

Refer back to the Taylor decomposition given in (25). Lemma 4 gives the asymptotic distribution of $\tilde{\gamma}_n(x_0)$, and the corresponding variance $\sigma_n^2(x_0)$. Lemma 5 provides a complementary result, controlling $\Delta_1(x)$.

Lemma 5. *Let $x_1, \ldots, x_n$ be independent and identically distributed on $[0,1]^d$. Suppose the conditional mean function $\mu(x) = \mathbb{E}[Y \mid X = x]$ is twice Lipschitz continuous, and let $\hat{\mu}_n(x_0)$ be the local linear forest estimate at x_0. Last, suppose we choose*

$$\lambda = \Theta(r_s^2\sqrt{s/n})$$

Then,

$$\Delta_1(x_0) = \mathcal{O}\left(n^{-\beta \cdot 0.78 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}} n^{(\beta-1)/4}\right)$$

3.3 Proof of Theorem 1

Recall the decomposition in equation (25),

$$\hat{\mu}(x_0) = \mu(x_0) + \Delta_1(x_0) + \hat{\gamma}_n(x_0).$$

Lemma 4 gives the distribution of $\tilde{\gamma}_n(x_0)$ for sufficiently large n, with asymptotic variance $\sigma_n^2(x_0)$. Corollary 3 establishes the coupling between $\hat{\gamma}_n(x_0)$ and $\tilde{\gamma}_n(x_0)$ at rate $\mathcal{O}_P(\lambda^{-1}r_s^2\sqrt{s/n})$. From these two results, we know there exists $\sigma_n(x_0) \to 0$ such that

$$\frac{\hat{\gamma}_n(x_0)}{\sigma_n(x_0)} \Rightarrow \mathcal{N}(0,1).$$

From Lemma 5, for any $\varepsilon > 0$,

$$\begin{aligned}\frac{\Delta_1(x_0)}{\sigma_n(x_0)} &= \mathcal{O}\left(n^{\frac{1}{2}\left(-\beta \cdot 0.78 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}\right)} n^{\frac{1}{2}(\beta-1)} n^{\frac{1}{2}(1+\varepsilon-\beta)}\right) \\ &= \mathcal{O}\left(n^{\frac{1}{2}\left(\varepsilon-\beta\left(0.78 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}\right)\right)}\right)\end{aligned}$$

Clearly for sufficiently small ε and $\beta > \beta_{\min}$, we have $\Delta_1(x)/\sigma_n(x_0) = o_p(1)$. Therefore, as $n, s \to \infty$ appropriately, Slutsky's Lemma implies that $(\hat{\mu}_n(x_0) - \mu(x_0)) \,/\, \sigma_n(x_0) \Rightarrow \mathcal{N}(0,\, 1)$.

4 Simulation Study

4.1 Methods

In this section, we compare local linear forests, (honest) random forests, and BART [Chipman, George, and McCulloch, 2010]. We also include a lasso-random forest baseline for local linear forests, proceeding as follows. Split the data and on the first half, train a lasso [Tibshirani, 1996] regression; on the second half, use a random forest to model the residuals. Like local linear forests, this method combines regression and forests and uses them to model different signals in the data, making it a natural comparison.

Local linear forests are tuned via 10-fold cross validation, with tuning parameters mtry (number of splitting variables), minimum leaf size, sample fraction used in each tree, and regularization for splitting and prediction. Variables for the regression at prediction are selected via the lasso. Random forests are run using `grf` [Tibshirani, Athey, Wager, Wright, and all contributors to the included version of Eigen, 2018] with honesty, and are cross-validated via the default cross-validation routine in `grf`, which selects values for mtry, minimum leaf size, sample fraction, and two parameters (alpha and imbalance penalty) that control split balance. Lasso is implemented via `glmnet` [Friedman, Hastie, and Tibshirani, 2010] and is cross-validated with their automatic cross-validation feature. The random forests we train on the lasso residuals are adaptive random forests cross-validated as before. Note that local linear regression is not included in these comparisons, since it is not feasible for $d > 6$; in the Appendix, we include Table 7, which compares local linear regression with this set of methods on lower dimensional linear models.

BART for treatment effect estimation is implemented following Hill [2011]. As is standard, we use the `BayesTree` package [Chipman and McCulloch, 2016] without any additional tuning. The motivation for not tuning is that, first, if we want to interpret the BART posterior in a Bayesian sense (as is often done), then cross-validating on the prior is hard to justify, and in fact most existing papers do not cross-validate BART. Second, BART is much slower than the other methods we consider, and cross-validation would make it prohibitively slow. For example, Table 2 displays the average runtime (in seconds) of this process for local linear forests (with fixed parameters and with cross-validation), random forests (with the same), lasso, and BART, on the example (9) with sample size $n = 600$, error standard deviation $\sigma = 1$, and dimension at $d = 5$ and $d = 50$. We can see that even without cross-validation, BART has by far the longest runtime of the peer methods.

4.2 Simulation Design

The first design we study is Friedman's example from equation (9). We generate $x_1, \ldots, x_n$ i.i.d. $U[0,1]^d$ and simulate responses

$$y = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10 x_4 + 5 x_5 + \epsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

We calculate root mean-squared error (RMSE) on 1000 test points, averaged over 50 runs. Figure 5 shows errors at $n = 600$ fixed, with dimension d varying from 10 to 50. There are two plots shown, to highlight the differences between error variance $\sigma = 5$

	Lasso-RF (cv)	BART	RF	RF (cv)	LLF	LLF (cv)
$d = 5$	1.94	34.6	0.39	1.59	0.59	4.05
$d = 50$	3.53	42.8	1.17	3.51	1.11	7.51

Table 2: Average runtime in seconds for each method on data generated from equation 9. In these simulations, to mirror the setup we use to evaluate RMSE, we fix training size $n = 600$, hold $\sigma = 5$ and and predict on 1000 test points. For local linear forests and random forests, we report both the runtime with pre-fixed parameters and the runtime with cross-validation (cv). Random forests with cross-validation are regression forests from `grf` with automatic self-tuning; both random forests included are honest. We report the runtime for each method averaged over 50 runs.

and $\sigma = 20$. Table 5 reports a grid of errors for dimensions 10, 30, and 50, with $n = 600$ and 1000, and σ taking values of 1, 5, 10, and 20.

The second design we consider is given in Section 1. We simulate $x_1, \ldots, x_n$ i.i.d. $U[-1, 1]^d$ and model responses as in equation (1),

$$y = \log(1 + \exp(6x_1)) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

Again we test on a grid, letting dimension d take values in 5, 10, and 20, n either 600 or 1000, and σ at 0.1, 1, and 2. We calculate RMSE over 50 runs on 1000 test points. Errors are reported in Table 6.

The third design is designed to test how local linear forests perform in a more adversarial setting, where we expect random forests to outperform. We simulate $x_1, \ldots, x_n$ i.i.d. $U[0, 1]^d$ and model responses as

$$y = \frac{10}{1 + \exp(-10 * (x_1 - 0.5))} + \frac{5}{1 + \exp(-10 * (x_2 - 0.5))} + \varepsilon, \quad \varepsilon \sim N(0, 5^2) \quad (30)$$

Here we test dimension $d = 5, 20$ and values of $n = 500, 2000$. For this simulation, we compare only random forests and local linear forests; we compute out of bag RMSE and average confidence interval coverage and length, all averaged over 50 runs. Results are reported in table 3.

4.3 Results

Figure 5 shows RMSE from equation 9 at $\sigma = 5$ (left) and $\sigma = 20$ (right). In Section 2, we showed that local linear forests and standard regression forests split on very different variables when generating weights. Our intuition is that these are splits we have saved; we model the strong linear effects at the end with the local regression, and use the forest splits to capture more nuanced local relationships for the weights. Local linear forests consistently perform well as we vary the parameters data-generating process, lending this credibility.

The lasso-random forest baseline lines up closely with random forests in the low-noise case; in the high-noise case, it aligns more closely with local linear forests, albeit with larger errors. BART, which does quite well in low-noise problems, suffers significantly when we decrease the signal-to-noise ratio as we do in this simulation. Note also that random forests suffer in the lower-noise case, but move to outperform in the

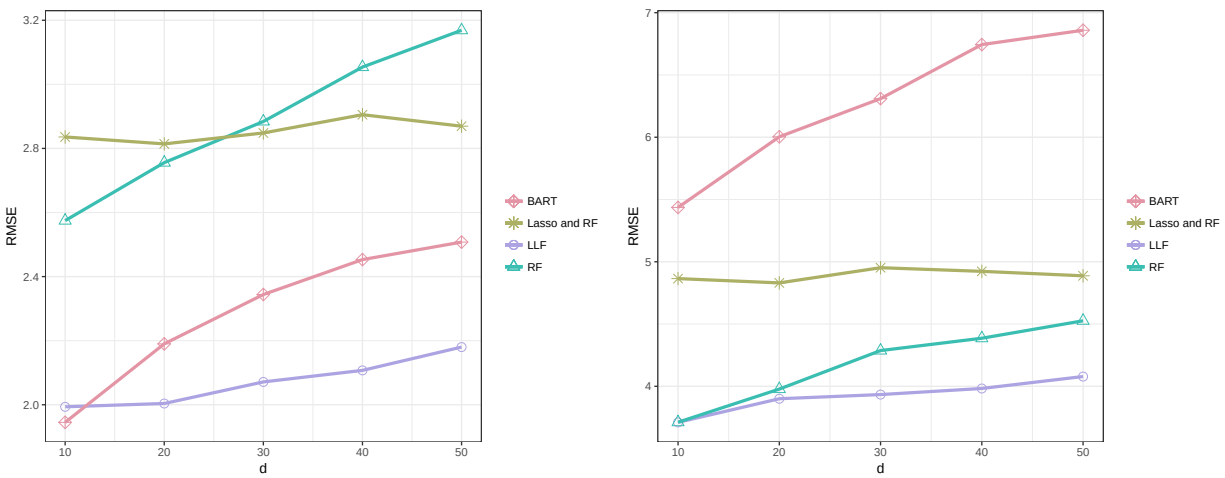

Figure 5: RMSE of predictions on 1000 *test samples from equation 9, with* $n = 600$ *held fixed and dimension* d *varied from 10 to 50. Plots are shown for error standard deviation* $\sigma = 5$ *(left) and* $\sigma = 20$ *(right). Error was calculated in increments of 10, and averaged over 50 runs per method at each step. Methods evaluated are (honest) random forests (RF), local linear forests (LLF), lasso and random forests, and BART.*

high noise case. As we only have $n = 600$ training points, and growing honest trees forces further subsampling, we have designed a difficult simulation for this method. Removing the honesty constraint and growing original random forests helps to improve RMSE, but adding the local linear correction lets us improve RMSE more while retaining honesty. Table 5 in Appendix A shows the fuller RMSE comparison from Friedman's model.

We move to the second simulation setup, equation 1, meant to evaluate how methods perform in cases with a strong linear trend in the mean. Tree-based methods will be prone to bias on this setup, as the forests cannot always split on x_1, and because the signal is global and smooth. Table 6 gives RMSE. Local linear forests do quite well here; they detect the strong linear signal in the tail, as we saw in Figure 1, and model it successfully throughout the range of the feature space. Full error results on the range of competing methods are given in the appendix in Table 6.

We also examine the behavior of our confidence intervals in the given simulation setups, shown here in Table 3. We give average coverage of 95% confidence intervals from random forests and local linear forests on the simulation setups in equations 1, 9, and 30, as well as average confidence interval length and RMSE. On equation 1, local linear forest confidence intervals are consistently shorter and closer to 95% coverage, with correspondingly lower mean squared error. Here, both random forests and local linear forests achieve fairly low RMSE and coverage at or above 88%. For the setup in equation 9, on the other hand, neither method achieves higher than 75% coverage, and the local linear forest confidence intervals are longer than the random forest confidence intervals. This is an encouraging result, indicating that local linear forests confidence intervals are more adaptable to the context of the problem; we would hope for long confidence intervals when detection is difficult.

We include the approximate step function in equation 30 to highlight a favorable

Setup	d	n	Coverage		Length		RMSE	
Equation 1			RF	LLF	RF	LLF	RF	LLF
	5	500	0.90	0.94	2.40	2.35	0.63	0.55
	5	2000	0.97	0.96	2.23	1.85	0.43	0.35
	20	500	0.88	0.92	2.23	2.13	0.68	0.55
	20	2000	0.89	0.96	2.14	2.12	0.17	0.09
Equation 9			RF	LLF	RF	LLF	RF	LLF
	5	500	0.54	0.65	3.56	3.82	2.36	2.03
	5	2000	0.63	0.69	3.17	3.21	1.77	1.58
	20	500	0.47	0.69	4.06	4.61	8.82	4.85
	20	2000	0.54	0.74	3.55	4.22	5.25	3.20
Equation 30			RF	LLF	RF	LLF	RF	LLF
	5	500	0.85	0.89	3.26	3.46	1.50	0.90
	5	2000	0.90	0.92	2.54	2.82	0.52	0.45
	20	500	0.85	0.89	3.36	3.19	1.50	0.98
	20	2000	0.90	0.90	2.71	2.36	0.62	0.46

Table 3: Average coverage and length of 95% *confidence intervals from honest random forests (RF) and local linear forests (LLF), along with RMSE on the same out of bag (OOB) predictions. OOB coverage is averaged over 50 runs of the simulation setups in equations 1, 9, and 30 and reported for the given values of dimension* d *and number of training points* n*. We hold* $\sigma = \sqrt{20}$ *constant for equation 1, and* $\sigma = 5$ *constant for equation 9, and train on sample fraction* 0.5*.*

example for random forests. Local linear forests see equivalent or better coverage on this setup, although at the cost of longer confidence intervals in low dimensions. Especially on small training datasets, local linear forests also improve on random forest predictions in RMSE.

4.4 Local Linear Causal Forests

In Section 1.1, we introduced a real-data example where the local linear extension of causal forests naturally applies. Evaluating errors empirically, however, is difficult, so we supplement that with a simulation also used by Wager and Athey [2018] in evaluating causal forests and Künzel, Sekhon, Bickel, and Yu [2017], used to evaluate their meta-learner called the X-learner. Here we let $X \sim U([0,1]^d)$. We fix the propensity $e(x) = 0.5$ and $\mu(x) = 0$, and generate a causal effect τ from each

$$\tau(x) = \zeta(x_1)\zeta(x_2), \quad \zeta(x) = \frac{2}{1 + \exp(-20(x - 1/3))} \tag{31}$$

$$\tau(x) = \zeta(x_1)\zeta(x_2), \quad \zeta(x) = 1 + \frac{1}{1 + \exp(-20(x - 1/3))} \tag{32}$$

We will assume unconfoundedness [Rosenbaum and Rubin, 1983]; therefore, because we hold propensity fixed, this is a randomized controlled trial.

We compare local linear forests, causal forests, and X-BART, which is the X-learner using BART as a base-learner. Causal forests as implemented by `grf` are tuned via

	Simulation 1 (equation 31)			Simulation 2 (equation 32)		
n	X-BART	CF	LLCF	X-BART	CF	LLCF
200	0.864	0.834	**0.673**	0.694	0.737	**0.601**
400	0.495	0.619	**0.471**	0.585	0.466	**0.420**
600	0.451	0.592	**0.423**	0.504	0.379	**0.337**
800	0.437	0.577	**0.412**	0.481	0.350	**0.331**
1000	0.354	0.463	**0.325**	0.480	0.347	**0.314**
1200	0.290	0.379	**0.280**	0.417	0.261	**0.257**

Table 4: Average RMSE of predicting the heterogeneous treatment effect τ_i on 100 repetitions of the simulation given in equation (31). We vary the sample size n from 200 to 1200 in increments of 200, always testing on 2000 test points. We report errors from local linear causal forests (LLCF), causal forests (CF), and the X-learner with BART as base learner (X-BART). Minimizing errors are reported in bold.

the automatic self-tuning feature, as were the regression forests used earlier in this simulation study. As in the prediction simulation studies, we do not cross-validate X-BART for two reasons. First, the authors recommend X-BART pecifically for when a user does not want to carefully tune, and second for timing as in Table 2. We acknowledge that this may hinder its performance . Local linear causal forests are tuned via cross-validation as described in 2.4. On these simulations, we

We consider relatively small sample sizes ranging from $n = 200$ to $n = 1200$ with dimension $d = 20$. The goal of this simulation is to evaluate how effectively we can learn a smooth heterogeneous treatment effect in the presence of many noise covariates. Wager and Athey [2018] emphasize equation 31 as a simulation that demonstrates how forests can suffer on the boundary of a feature space, because there is a spike near $x = 0$. RMSE over 100 repetitions is reported in Table 4. We can see that in these simulations, local linear forests give a significant improvement over causal forests. Both of these setups are reasonable tests for how a method can learn heterogeneity, and demonstrate potential for meaningful improvement with thoughtful variable selection and robustness to smooth heterogeneous signals.

4.5 The Value of Local Linear Splitting

We close this section with an experiment that highlights the benefit of the splitting rule proposed in Section 2.1. We generate $x_1, \ldots, x_n$ independently and uniformly over $[0, 1]^d$. We hold a cubic signal $20(x_1 - 0.5)^3$ constant across simulations, and on each run increase the dimension and add another linear signal. Formally, we let $\xi_j = \mathbb{1}\{d \leq j\}$ and generate responses

$$y = 20(x_1 - 0.5)^3 \xi_1 + \sum_{j=2}^{3} 10 x_j \xi_j + \sum_{j=4}^{5} 5 x_j \xi_j + \sum_{j=6}^{20} 2 x_j \xi_j \tag{33}$$

For example, at simulation 3 we have $\xi_1, \xi_2, \xi_3 = 1$ and hence we model $y = 20(x_1 - 0.5)^3 + 10x_2 + 10x_3$. RMSE is displayed in Figure 6.

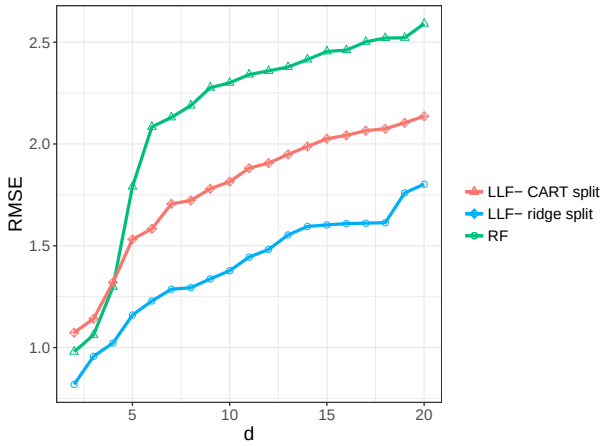

Figure 6: Results from testing different splitting rules on data generated from equation 33. Here the x-axis is dimension d, varying from 2 to 20, and we plot the RMSE of prediction from random forests and from local linear forests with CART splits and with the ridge residual splits. We let $n = 600$ *and check results on* 600 *test points at* 50 *runs for each value of* d.

In low dimension and with few linear signals, all three methods are comparable. However, they begin to differ quickly. Random forests are not designed for models with so many global linear signals, and hence their RMSE increases dramatically with d. Moreover, as we add more linear effects, the gap between the two candidate splitting rules grows; heuristically, it becomes more important not to waste splits, and the residual splitting rule gives greater improvements. Note that at a certain point, however, the gap between splitting rules stays constant. Once the forests simply cannot fit a more complex linear function with a fixed amount of data, the marginal benefits of the residual splitting rule level out. We show this to emphasize the contexts in which this splitting rule meaningfully affects the results.

5 Discussion

In this paper, we proposed local linear forests as a modification of random forests equipped to model smooth signals and fix boundary bias issues. We presented asymptotic theory showing that, if we can assume smoother signals, we can get better rates of convergence than standard random forests. We showed on the welfare dataset that local linear forests can effectively model smooth heterogeneous causal effects, and illustrated in several simulations when and why they out-perform competing methods. We also gave confidence intervals from the delta method for the regression prediction case, and demonstrated their effectiveness in simulations.

The regression adjustments in local linear forests prove especially useful when some covariates have strong global effects with moderate curvature. Furthermore, the local regression adjustment provides centered predictions, adjusting for errors due to an asymmetric set of neighbors. Relative to the lasso, local linear forests provide more scale-invariance and relatively little hand-tuning. Likely there is a useful polynomial basis corresponding to every situation where local linear forests perform well, but

this requires hand-tuning the functional form for competitive performance, and is not automatically suited to a mix of smooth and non-smooth signals. For a departure from frequentist techniques, BART and Gaussian processes are both hierarchical Bayesian methods; BART can be viewed as a form of Gaussian process with a flexible prior, making BART the preferred baseline.

There remains room for meaningful future work on this topic. In some applications, we may be interested in estimating the slope parameter $\theta(x)$, rather than merely accounting for it to improve the precision of $\mu(x)$. While local linear forests may be an appropriate method for doing so, we have not yet explored this topic and think it could be of significant interest.

Acknowledgments

R.F. was supported by the DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. S.A. was supported by the Sloan Foundation and ONR grant N00014-17-1-2131. S.W. was supported by a Facebook Faculty award. The authors would like to thank Guido Imbens and Steve Yadlowsky for helpful comments.

References

Alberto Abadie and Guido W Imbens. Bias-corrected matching estimators for average treatment effects. *Journal of Business & Economic Statistics*, 29(1):1–11, 2011.

Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Comput.*, 9(7):1545–1588, October 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.7.1545.

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

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *Annals of Statistics*, forthcoming, 2018.

Sumanta Basu, Karl Kumbier, James B. Brown, and Bin Yu. Iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences*, 2018. ISSN 0027-8424. doi: 10.1073/pnas.1711236115. URL http://www.pnas.org/content/early/2018/01/17/1711236115.

Gérard Biau. Analysis of a random forests model. *J. Mach. Learn. Res.*, 13(1):1063–1095, April 2012. ISSN 1532-4435.

Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *JMLR*, 9:2015–2033, 2008.

Adam Bloniarz, Ameet Talwalkar, Bin Yu, and Christopher Wu. Supervised neighborhoods for distributed nonparametric regression. In *Artificial Intelligence and Statistics*, pages 1450–1459, 2016.

L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN 9780412048418.

Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324.

Peter Bühlmann and Bin Yu. Analyzing bagging. *The Annals of Statistics*, 30(4): 927–961, 2002.

Ming-Yen Cheng, Jianqing Fan, and J. S. Marron. On automatic boundary corrections. *The Annals of Statistics*, 25(4):1691–1708, 1997. ISSN 00905364.

Hugh Chipman and Robert McCulloch. *BayesTree: Bayesian Additive Regression Trees*, 2016. URL https://CRAN.R-project.org/package=BayesTree. R package version 0.3-1.4.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bart: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4(1):266–298, 03 2010. doi: 10.1214/09-AOAS285.

William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

William S. Cleveland and Susan J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988. doi: 10.1080/01621459.1988.10478639.

D Richard Cutler, Thomas C Edwards Jr, Karen H Beard, Adele Cutler, Kyle T Hess, Jacob Gibson, and Joshua J Lawler. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007.

Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam, 1982.

Bradley Efron and Charles Stein. The jackknife estimate of variance. *The Annals of Statistics*, pages 586–596, 1981.

Jianqing Fan and Irene Gijbels. Variable bandwidth and local linear regression smoothers. *Ann. Statist.*, 20(4):2008–2036, 12 1992. doi: 10.1214/aos/1176348900.

Jianqing Fan and Irne Gijbels. *Local polynomial modelling and its applications*. Number 66 in Monographs on statistics and applied probability series. Chapman & Hall, London [u.a.], 1996. ISBN 0412983214.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33 (1):1–22, 2010. URL http://www.jstatsoft.org/v33/i01/.

Jerome H. Friedman. Multivariate adaptive regression splines. *Ann. Statist.*, 19(1): 1–67, 03 1991. doi: 10.1214/aos/1176347963.

João Gama. Functional trees. *Mach. Learn.*, 55(3):219–250, June 2004. ISSN 0885-6125. doi: 10.1023/B:MACH.0000027782.67192.13. URL http://dx.doi.org/10.1023/B:MACH.0000027782.67192.13.

Donald P. Green and Holger L. Kern. Modeling heterogeneous treatment effects in survey experiments with bayesian additive regression trees. *Public Opinion Quarterly*, 76(3):491–511, 2012. doi: 10.1093/poq/nfs036.

James J Heckman, Hidehiko Ichimura, and Petra Todd. Matching as an econometric evaluation estimator. *The review of economic studies*, 65(2):261–294, 1998.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, March 1963.

Torsten Hothorn, Berthold Lausen, Axel Benner, and Martin Radespiel-Troger. Bagging survival trees. *Statistics in Medicine*, 23:77–91, 2004.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Aram Karalič. Employing linear regression in regression tree leaves. In *Proceedings of the 10th European Conference on Artificial Intelligence*, ECAI '92, pages 440–441, New York, NY, USA, 1992. John Wiley & Sons, Inc. ISBN 0-471-93608-1. URL http://dl.acm.org/citation.cfm?id=145448.146775.

Soren R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Meta-learners for Estimating Heterogeneous Treatment Effects using Machine Learning. *ArXiv e-prints*, June 2017.

Yehua Li and Tailen Hsing. Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *Ann. Statist.*, 38 (6):3321–3351, 12 2010. doi: 10.1214/10-AOS813.

Catherine Loader. *locfit: Local Regression, Likelihood and Density Estimation.*, 2013. URL https://CRAN.R-project.org/package=locfit. R package version 1.5-9.1.

Clive Loader. *Local regression and likelihood*. New York: Springer-Verlag, 1999. ISBN 0-387-9877.

Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999, 2006.

Lucas Mentch and Giles Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *J. Mach. Learn. Res.*, 17(1):841–881, January 2016. ISSN 1532-4435.

Whitney K. Newey. Kernel estimation of partial means and a general variance estimator. *Econometric Theory*, 10(2):233–253, 1994. ISSN 02664666, 14694360.

Xinkun Nie and Stefan Wager. Learning objectives for treatment effect estimation. *arXiv preprint arXiv:1712.04912*, 2017.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL https://www.R-project.org/.

Peter M Robinson. Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 931–954, 1988.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983. doi: 10.1093/biomet/70.1.41.

Thomas Rusch and Achim Zeileis. Gaining insight with recursive partitioning of generalized linear models. *Journal of Statistical Computation and Simulation*, 83(7):1301–1315, 2013. doi: 10.1080/00949655.2012.658804. URL https://doi.org/10.1080/00949655.2012.658804.

Erwan Scornet, Grard Biau, and Jean-Philippe Vert. Consistency of random forests. *Ann. Statist.*, 43(4):1716–1741, 08 2015. doi: 10.1214/15-AOS1321.

Joseph Sexton and Petter Laake. Standard errors for bagged and random forest estimators. *Computational Statistics & Data Analysis*, 53(3):801–811, 2009.

Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–620, 1977.

Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43(6):1947–1958, 2003.

M. Taddy, C.-S. Chen, J. Yu, and M. Wyle. Bayesian and empirical Bayesian forests. *ArXiv e-prints*, feb 2015.

Julie Tibshirani, Susan Athey, Stefan Wager, Marvin Wright, and all contributors to the included version of Eigen. *grf: Generalized Random Forests (Beta)*, 2018. URL https://github.com/swager/grf. R package version 0.9.3.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

Robert Tibshirani and Trevor Hastie. Local likelihood estimation. *Journal of the American Statistical Association*, 82(398):559–567, 1987.

Luís Torgo. Functional models for regression tree leaves. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 385–393, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3. URL http://dl.acm.org/citation.cfm?id=645526.657280.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113 (523):1228–1242, 2018.

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

Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *J. Mach. Learn. Res.*, 15(1): 1625–1651, January 2014. ISSN 1532-4435.

Fang Yao, Hans-Georg Muller, and Jane-Ling Wang. Functional linear regression analysis for longitudinal data. *Ann. Statist.*, 33(6):2873–2903, 12 2005. doi: 10.1214/009053605000000660.

Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, 2008. doi: 10.1198/106186008X319331. URL https://doi.org/10.1198/106186008X319331.

Yichen Zhou and Giles Hooker. Boulevard: Regularized stochastic gradient boosted trees and their limiting distribution. *arXiv preprint arXiv:1806.09762*, 2018.

Ruoqing Zhu, Donglin Zeng, and Michael R Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.

Appendix A: Remaining Simulation Results

We include first Table 5, giving a full error comparison of the lasso-random forest baseline, BART, random forests, and local linear forests, on equation 9. Errors are reported on dimension ranging from 5 to 20, σ from 1 to 20, and $n = 600$ and 1000, averaged over 50 training runs.

p	n	σ	Lasso-RF	BART	RF	LLF
10	600	1	1.69	**0.82**	2.21	1.64
10	1000	1	1.45	**0.65**	1.95	1.35
30	600	1	1.91	**0.99**	2.49	1.68
30	1000	1	1.64	**0.78**	2.19	1.57
50	600	1	2.00	**1.21**	2.62	1.85
50	1000	1	1.70	**0.84**	2.48	1.71
10	600	5	2.17	**1.92**	2.54	1.93
10	1000	5	1.96	**1.70**	2.31	1.80
30	600	5	2.43	2.44	2.86	**2.02**
30	1000	5	2.25	2.01	2.57	**1.92**
50	600	5	2.53	2.51	3.12	**2.08**
50	1000	5	2.34	2.24	2.79	**2.04**
10	600	10	2.62	3.04	2.96	**2.56**
10	1000	10	2.32	2.78	2.74	**2.28**
30	600	10	3.01	3.60	3.49	**2.62**
30	1000	10	2.64	3.42	2.95	**2.35**
50	600	10	3.24	3.87	3.78	**2.65**
50	1000	10	2.82	3.57	3.50	**2.36**
10	600	20	**3.67**	5.34	3.72	3.71
10	1000	20	**3.18**	4.92	3.29	3.46
30	600	20	4.11	5.92	4.28	**3.93**
30	1000	20	3.62	5.79	4.15	**3.57**
50	600	20	4.26	6.67	4.52	**4.08**
50	1000	20	3.87	6.19	4.60	**3.80**

Table 5: RMSE from simulations on equation 9. We vary the dimension p from 10 to 50 predictors in increments of 20, and consider error standard deviation σ ranging from 1 to 20, for a variety of signal-to-noise ratios. Note that for this setting, $\mathrm{Var}(\mathbb{E}[Y \mid X]) \approx 23.8$*, as approximated over 10,000 Monte Carlo repetitions; so letting $\sigma = 1$ corresponds to a signal-to-noise ratio of about 23.8, while letting $\sigma = 20$ corresponds to a signal-to-noise ratio of about* 0.24*. We train on $n = 600$ and $n = 1000$ points, and report test errors from predicting on* 1000 *test points. All errors reported are averaged over 50 runs and the methods are cross-validated as described in Section 4.1. Minimizing errors are reported in bold.*

We include next Table 6, again giving a more complete error comparison of the lasso-random forest baseline, BART, random forests, and local linear forests, on equation 1. Errors are reported on dimension ranging from 5 to 20, σ from 0.1 to 2, and $n = 600$ and 1000, averaged over 50 training runs.

To close this section, we consider some basic linear and polynomial models in low dimensions, in order to effectively compare local linear forests with local linear regres-

d	n	σ	Lasso-RF	BART	RF	LLF
5	600	0.1	0.10	0.07	0.10	**0.05**
5	1000	0.1	0.07	0.06	0.09	**0.05**
10	600	0.1	0.13	0.08	0.11	**0.07**
10	1000	0.1	0.09	**0.06**	0.13	0.09
20	600	0.1	0.14	**0.08**	0.10	0.09
20	1000	0.1	0.10	**0.06**	0.13	0.10
5	600	1	0.27	0.27	0.21	**0.16**
5	1000	1	0.22	0.25	0.18	**0.14**
10	600	1	0.30	0.36	0.23	**0.16**
10	1000	1	0.26	0.32	0.21	**0.15**
20	600	1	0.34	0.41	0.22	**0.16**
20	1000	1	0.27	0.35	0.21	**0.15**
5	600	2	0.45	0.53	0.32	**0.27**
5	1000	2	0.39	0.50	0.29	**0.23**
10	600	2	0.51	0.61	0.35	**0.27**
10	1000	2	0.42	0.55	0.30	**0.24**
20	600	2	0.56	0.69	0.38	**0.27**
20	1000	2	0.45	0.61	0.31	**0.24**

Table 6: RMSE from simulations on equation 1 on local linear forests, random forests, lasso-random forest, and BART. We vary sample size n, error variance σ, and ambient dimension p, and report test error on 1000 *test points. We estimate* $\mathrm{Var}[\mathbb{E}[Y \mid X]]$ *as* 3.52 *over 10,000 Monte Carlo repetitions, so that signal-to-noise ratio ranges from* 352 *at* $\sigma = 0.1$ *to* 0.88 *at* $\sigma = 2$*. All errors are averaged over 50 runs, and minimizing errors are in bold.*

sion. We simulate $X \sim U[0,1]^3$ and model responses from two models,

$$y = 10x_1 + 5x_2 + x_3 + \varepsilon \tag{34}$$

$$y = 10x_1 + 5x_2^2 + x_3^3 + \varepsilon, \tag{35}$$

where $\varepsilon \sim N(0, \sigma^2)$ and $\sigma \in \{1, 5, 10\}$. RMSE on the truth is reported, averaged over 50 runs, for lasso, local linear regression, BART, random forests, adaptive random forests, and local linear forests. In the simple linear case in equation 34, we see that lasso outperforms the other methods, as we would expect; in the polynomial given in equation 35, local linear regression performs the best, followed by BART ($\sigma = 1$ case) and local linear forests ($\sigma = 5, 10$ cases).

Setup	σ	Lasso	LLR	BART	RF	LLF
Equation 34	1	**0.12**	0.15	0.48	0.73	0.22
	5	**0.39**	0.92	1.27	1.25	0.96
	10	**0.70**	1.70	2.37	1.76	1.56
Equation 35	1	1.55	**0.22**	0.50	0.86	0.69
	5	1.55	**0.92**	1.31	1.32	1.28
	10	1.66	**1.44**	1.83	1.70	1.68

Table 7: RMSE from simulations on equations 34 and 35 on lasso, local linear regression (LLR), BART, random forests, adaptive random forests, and local linear forests. We vary error variance σ from 1 to 10 and fix $n = 600, d = 3$. All errors are averaged over 50 runs, and minimizing errors are in bold.

Appendix B: Remaining Proofs

Here we give the proofs remaining from Section 3. We begin with Lemma 6, which gives a bound on the bias of $\hat{\gamma}_n(x_0)$.

Coupling between $\hat{\gamma}_n(x_0)$ and $\tilde{\gamma}_n(x_0)$

Lemma 6. *Suppose we have training data $(x_1, Y_1), \ldots, (x_n, Y_n)$ with $x_1, \ldots, x_n$ i.i.d. $[0,1]^d$ and $\gamma(x) = \mathbb{E}[Y \mid X = x]$ twice Lipschitz continuous. Assume further that we have a forest grown on honest, regular trees T, and all conditions of Lemma 2 from Wager and Athey [2018] hold. Then, for $\omega \leq 0.2$, the bias of the random forest prediction $\hat{\gamma}(x_0)$ at test point x_0 is bounded by*

$$|\mathbb{E}[\hat{\gamma}(x_0)] - \gamma(x_0)| = \mathcal{O}\left(s^{-0.78 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})} \frac{\pi}{d}}\right)$$

Proof. We begin with two observations. First by assuming the conditional mean function is twice Lipschitz continuous, and second by honesty, we have

$$|\mathbb{E}[\tilde{Y} \mid X \in L(x_0)] - \mathbb{E}[\tilde{Y} \mid X = x_0]| \leq C \mathrm{diam}^2(L(x_0))$$
$$\mathbb{E}[T(x_0) - \mathbb{E}[\tilde{Y} \mid X = x_0] = \mathbb{E}[\mathbb{E}[\tilde{Y} \mid X \in L(x_0)] - \mathbb{E}[\tilde{Y} \mid X = x_0]]$$

Let $\eta = 1.25\sqrt{\log((1-\omega)^{-1})} \leq 0.6$. Then $1.98(1-\eta) \geq 0.79$, and $\eta^2/2 \geq 0.78 \log((1-\omega)^{-1})$. Equation 29 gives

$$\mathbb{P}\left(\mathrm{diam}^2(L(x_0)) \geq \sqrt{d}\left(\frac{s}{2k+1}\right)^{-0.79 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})} \frac{\pi}{d}}\right) \leq d\left(\frac{s}{2k+1}\right)^{-0.78 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})} \frac{\pi}{d}}$$

The proof of Theorem 3 in Wager and Athey [2018] establishes

$$|\mathbb{E}[T(x_0)] - \mathbb{E}[\tilde{Y} \mid X = x_0]| \lesssim d\left(\frac{s}{2k-1}\right)^{-0.78 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})} \frac{\pi}{d}} \times \mathcal{O}(1)$$

Since a forest is an average of trees, this bound extends to the bias of the forest. □

The following two results use the theory of U-statistics to complete the coupling between $\hat{\gamma}_n(x_0)$ and $\tilde{\gamma}_n(x_0)$, which relies on appropriately bounding M_λ.

Proof of Lemma 2

Let $x_1, \ldots, x_n$ be independent random variables, and recall that a U-statistic takes the form

$$U = \frac{1}{n^{(s)}} \sum_{n,s} g(x_{i_1}, \ldots, x_{i_s}),$$

where $\sum_{n,s}$ is over all s-tuples and $n^{(s)} = n!/(n-s)!$. Writing M_λ as

$$M_\lambda = \frac{1}{n^{(s)}} \sum_{n,s} \sum_{i=1}^{s} \alpha_i \begin{pmatrix} 1 \\ x_i - x \end{pmatrix}^{\otimes 2} + \begin{pmatrix} 0 \\ \lambda \end{pmatrix},$$

we note that every entry of M_λ is a one-dimensional U-statistic. Suppose we have a, b such that $a \leq g(x_1, \ldots, x_s) \leq b$. Then, Hoeffding [1963], gives

$$\mathbb{P}(U - \mathbb{E}[U] \geq t) \leq \exp\left(-2\frac{n}{s}t^2/(b-a)^2\right)$$

Let U be any entry of M_λ. Then for any $x_1, \ldots, x_s$, with high probability given in 29 we have $-r_s^2 \leq g(x_1, \ldots, x_s) \leq r_s^2$. Hence we may bound $P(U - \mathbb{E}[U] \geq t)$ as

$$P(U - \mathbb{E}[U] \geq t) \leq \exp\left(-\frac{n}{s}t^2/2r_s^4\right)$$

Therefore, across all elements of M_λ,

$$||M_\lambda - \mathbb{E}[M_\lambda]||_\infty = \mathcal{O}_P\left(\sqrt{s/n}r_s^2\right)$$
$$||M_\lambda^{-1} - \mathbb{E}[M_\lambda]^{-1}||_\infty = \mathcal{O}_P\left(\lambda^{-1}\sqrt{s/n}r_s^2\right)$$

□

Proof of Corollary 3

From Lemma 2, it follows that

$$||M_\lambda^{-1} - \mathbb{E}[M_\lambda]^{-1}||_\infty = \mathcal{O}\left(\lambda^{-1}r_s^2\sqrt{s/n}\right)$$

Clearly $\sum_{i=1}^n \alpha_i \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \varepsilon_i = \mathcal{O}(1)$, so we have

$$\sum_{i=1}^n \alpha_i M_\lambda^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \varepsilon_i = \sum_{i=1}^n \alpha_i \mathbb{E}[M_\lambda]^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \varepsilon_i + \mathcal{O}\left(\lambda^{-1}r_s^2\sqrt{s/n}\right)\mathcal{O}(1)$$

□

Note that this immediately implies we must choose $\lambda = \mathcal{O}(r_s^2\sqrt{s/n})$.

Proof of Lemma 4

Proving this lemma entails a study of $\tilde{\gamma}_n(x_0)$. We begin by detailing how honest trees operate under the relevant dependence setup, and then give a central limit theorem at the appropriate subsampling rate.

Recall we have data $(x_1, Y_1), \ldots, (x_n, Y_n)$, where $Y_i = \mu(x_i) + \varepsilon_i$ and a test point x_i. Recall the notation M_λ given in (14). Let

$$S_i = \mathbb{1}\{x_i \in L(x_0, T)\}/|\{L(x_0, T)\}|$$

and recall the definition

$$\tilde{Y}_i = e_1^T \mathbb{E}[M_\lambda]^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \varepsilon_i.$$

Then predictions from a tree T are $\sum_{i=1}^n S_i \tilde{Y}_i$. We want to establish that

$$\mathbb{E}[T(x_0) \mid x_1] = \mathbb{E}[S_1 \mid x_1]\mathbb{E}[Y_1 \mid x_1] \tag{36}$$

In the proof of Theorem 5 from Wager and Athey [2018], honesty automatically provides $\mathbb{E}[S_1 \mid (x_1, Y_1)] = \mathbb{E}[S_1 \mid x_1]$, giving (36) immediately. Now, we establish that the relationships shown in Wager and Athey [2018], previously guaranteed by honesty and conditional independence, still hold without independence as long as we have zero correlation. It is sufficient to examine the behavior of one tree T on this problem.

Let us expand the conditional expectation of the tree predictions given x_1 (without loss of generality).

$$\mathbb{E}[T(x_0) \mid x_1] = \mathbb{E}\left[\sum_{i=1}^s S_i \tilde{Y}_i \mid x_1\right] = \mathbb{E}[S_1 \tilde{Y}_1 \mid x_1] + \sum_{i=2}^s \mathbb{E}[S_i \tilde{Y}_i \mid x_1]$$

While $\mathbb{E}[M_\lambda]$ and $\alpha_i(x_0)$ are still not independent, they are crucially uncorrelated given x_1. Define

$$H_i = S_i e_1^T \mathbb{E}[M_\lambda]^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix}.$$

Consider first the summation term $\sum_{i=2}^s \mathbb{E}[S_i \tilde{Y}_i \mid x_1]$. By construction, $\tilde{S}_2 \tilde{Y}_2 = H_2 \varepsilon_2$; and so

$$\begin{aligned} \mathbb{E}[S_2 \tilde{Y}_2 \mid x_1] &= \mathbb{E}[H_2 \varepsilon_2 \mid x_1] \\ &= \mathbb{E}[H_2 \mid x_1]\mathbb{E}[\varepsilon_2 \mid x_1] \\ &= \mathbb{E}[H_2 \mid x_1]\mathbb{E}[\varepsilon_2] = 0. \end{aligned}$$

Because S_1, Y_1 are uncorrelated given x_1, we indeed have $\mathbb{E}[S_1 \tilde{Y}_1 \mid x_1] = \mathbb{E}[S_1 \mid x_1]\mathbb{E}[\tilde{Y}_1 \mid x_1]$. Combining these observations gives Equation (36). Last, we can condition on (x_1, Y_1) and achieve an analogous result through the same steps. Proposition 7 gives a corresponding lower bound on $\mathrm{Var}(T)$. Therefore, Theorem 8 of Wager and Athey [2018] establishes the existence of $\sigma_n(x_0) \to 0$ such that

$$\frac{\tilde{\gamma}_n(x_0) - \mathbb{E}[\tilde{\gamma}_n(x_0)]}{\sigma_n(x_0)} \Rightarrow \mathcal{N}(0, 1)$$

We need to show that we can replace $\mathbb{E}[\tilde{\gamma}_n(x_0)]$ by $\gamma(x_0) = 0$. Note that any continuity conditions applied to $\mu(x)$ must immediately apply to $\hat{\gamma}_n(x_0)$. Therefore, Lemma 6 applies. Since $\tilde{\gamma}_n(x_0)$ is coupled with $\hat{\gamma}_n(x_0)$, the bias of a tree, and hence of a forest, is

$$|\mathbb{E}[\tilde{\gamma}_n(x_0)]| = \mathcal{O}\left(n^{-\beta * 0.78 \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}}\right)$$

For any $\varepsilon > 0$, Wager and Athey [2018] give the following bound, where k is the minimum leaf size and C is a constant.

$$\sigma_n^2(x_0) \gtrsim \frac{C}{2k}\frac{s}{n}\frac{\mathrm{Var}(Y \mid X = x_0)}{\log(s)^d} = \Omega(n^{\beta-1-\varepsilon})$$

Therefore, $\mathbb{E}[\tilde{\gamma}_n(x_0)]/\sigma_n(x_0)$ converges to 0 for sufficiently small $\varepsilon > 0$, as long as

$$\beta > 1 - \left(1 + \frac{1}{1.56}\frac{\log(\omega^{-1})}{\log((1-\omega)^{-1})}\frac{d}{\pi}\right)^{-1} = \beta_{\min}$$

□

Proposition 7. *Suppose the conditions of Lemma 4 hold, and moreover that*

$$\left(\mathbb{E}\left[\sum_{i=1}^{n} H_i\right]\right)^2 \geq \mathcal{O}(1),$$

for H_i as given in the proof of Lemma 4. Moreover, assume that the breakdown $S_i\tilde{Y}_i = H_i\varepsilon_i$ corresponds to $\varepsilon_i = Var(\tilde{Y}_i \mid x_i)$. Last, suppose $\mathrm{Var}(\tilde{Y}_i \mid x_i)$ is equal for all values of i. Then, we have the following bound on $\mathrm{Var}(T)$.

$$\mathrm{Var}(T) \gtrsim \frac{\mathrm{Var}(\tilde{Y} \mid X = x_0)}{2k}$$

Proof. We proceed by explicitly computing $\mathrm{Var}(T)$.

$$\mathrm{Var}(T) = \mathrm{Var}\left(\sum_{i=1}^{n} H_i\varepsilon_i\right) = \sum_{i=1}^{n}\mathrm{Var}(H_i\varepsilon_i) + 2\sum_{i=1}^{n}\sum_{j<i}\mathrm{Cov}(H_i\varepsilon_i, H_j\varepsilon_j)$$

First, observe that $\mathbb{E}[H_i\varepsilon_i] = 0$, thus $\mathrm{Var}(H_i\varepsilon_i) = \mathbb{E}[H_i^2\varepsilon_i^2]$. One can similarly check that $E[H_i^2\varepsilon_i^2] = \mathbb{E}[\mathbb{E}[H_i^2 \mid x_i]\mathbb{E}[\varepsilon_i^2]] = \mathbb{E}[H_i^2]\,\mathrm{Var}(\varepsilon_i)$. From the Proposition statement we have assumed that $\mathrm{Var}(\varepsilon_i) = \mathrm{Var}(\tilde{Y}_i \mid x_i) = \mathrm{Var}(\tilde{Y} \mid X = x_0)$. Hence,

$$\mathrm{Var}(T) = \sum_{i=1}^{n}\mathrm{Var}(\tilde{Y} \mid X = x_0)\mathbb{E}[H_i^2] + 2\sum_{i=1}^{n}\sum_{j<i}\mathrm{Cov}(H_i\varepsilon_i, H_j\varepsilon_j)$$

Quick algebra verifies that $\sum_{i=1}^{n}\sum_{j<i}\mathrm{Cov}(H_i\varepsilon_i, H_j\varepsilon_j) = 0$. Furthermore, recall that $H_i = 0$ if $S_i = 0$, so $\sum_{i=1}^{n} H_i$ is a sum of $|\{i : x_i \in L(x_0)\}|$ nonzero terms. The Cauchy-Schwartz inequality then gives

$$\sum_{i=1}^{n}\mathbb{E}[H_i^2] \geq \frac{1}{|\{i : x_i \in L(x_0)\}|}\left(\sum_{i=1}^{n}\mathbb{E}[H_i]\right)^2$$

Recall that $\text{Var}(T) = \text{Var}\left(\sum_{i=1}^n H_i \varepsilon_i\right)$. Expanding to $\text{Var}(\tilde{Y} \mid X = x_0) \sum_{i=1}^n \mathbb{E}[H_i^2]$, and noting the assumed bound on $\left(\mathbb{E}\left[\sum_{i=1}^n H_i\right]\right)^2$,

$$\begin{aligned}\text{Var}(T) &\geq \frac{\text{Var}(\tilde{Y} \mid X = x_0)}{|\{i : x_i \in L(x_0)\}|} \left(\mathbb{E}\left[\sum_{i=1}^n H_i\right]\right)^2 \\ &\gtrsim \frac{\text{Var}(\tilde{Y} \mid X = x_0)}{2k}.\end{aligned}$$

This establishes the necessary lower bound on $\text{Var}(T)$. □

Proof of Lemma 5

Recall from (25) that we can decompose $\hat{\mu}_n(x_0)$ with dominating bias term

$$\Delta_1(x_0) = \sum_{i=1}^n e_1^T M_\lambda^{-1} \begin{pmatrix} 1 \\ x_i - x_0 \end{pmatrix} \alpha_i \nabla\mu(x_0) \begin{pmatrix} 0 \\ x_i - x_0 \end{pmatrix}^T$$

Define the weighted average and corresponding centered matrix

$$\begin{aligned}\bar{\mathbf{X}} &:= \sum_{i=1}^n \alpha_i x_i \\ x_C &:= x - \bar{\mathbf{X}}\end{aligned}$$

Then write $\Delta_1(x)$ as a function of a vector ν, where

$$\Delta_1(\nu) = x_C(I - (x_C^T A x_C + \lambda I)^{-1} x_C^T A x_C)\nu,$$

for $\nu = \begin{pmatrix} 0 & \nabla\mu(x) \end{pmatrix}^T$. Last, define

$$B = I - (x_C^T A x_C + \lambda I)^{-1} x_C^T A x_C, \tag{37}$$

so that $\Delta_1(\nu) = X_C B \nu$. We must bound $\Delta_1(\nu)$ in probability, and hence can restrict to $||\nu||_2 \leq 1$. Moreover, we know $||X_C||_2 \leq \mathcal{O}(r_s)$, where r_s is given in equation 29. Recall the definition of the matrix operator norm

$$||B||_{op} = \inf\{c \geq 0 : ||B\nu||_2 \leq c||\nu||_2 \text{ for all } v \in \mathbb{R}^{p+1}\} = ||B||_*, \tag{38}$$

where $||B||_*$ is the largest eigenvalue of $B^T B$. By this definition, we can clearly bound

$$\sup_{v:||v||_2 \leq 1} ||\Delta_1(\nu)||_2 \leq ||B||_* \, \mathcal{O}(r_s). \tag{39}$$

Let $M = x_C^T A x_C$ and write the operator matrix B from (37) as

$$B = I - (M + \lambda I)^{-1} M \tag{40}$$

First, suppose $\lambda = 0$. Then $B = I - M^{-1}M = I$, and corresponding first-order error is $\Delta_1(\nu) = \mathbf{X}_C(I - I)\nu = 0$. Therefore any probability bound for nonzero λ will

trivially hold for this case. More importantly, note this intuition; if $\lambda = 0$, we do not apply a ridge correction, and hence do not incur the subsequent first-order bias.

Let σ_i be the eigenvalues of M. For nonzero λ, basic linear algebra verifies that the eigenvalues of $(M + \lambda I)^{-1}M$ are $\frac{\sigma_i}{\sigma_i+\lambda}$, and hence that the eigenvalues of B are

$$\frac{\lambda}{\lambda + \sigma_i} \tag{41}$$

Therefore,

$$||B||_* = \max_{\sigma_i} \left\{ \frac{\lambda}{\lambda + \sigma_i} \right\}$$

Certainly $\frac{\lambda}{\lambda+\sigma_i}$ is maximized at the smallest value of σ_i, which corresponds to the smallest eigenvalue of $M = \mathbf{X}_C^T A x_C$, and the inverse of the largest eigenvalue of M^{-1}. That is,

$$||B||_* = \frac{\lambda}{\lambda + (||M^{-1}||_*)^{-1}}$$

Recall that we assumed $\lambda = \mathcal{O}(r_s^2\sqrt{s/n})$. As $||M^{-1}||_* = ||(x_C^T A x_C)^{-1}||_* = \mathcal{O}(1/r^2)$, we have $||B||_* = \mathcal{O}(\lambda r_s/(\lambda + r_s^2)) = \mathcal{O}(\lambda/r_s)$. By our choice of λ, we have $\mathcal{O}\left(\lambda/r_s\right) = \mathcal{O}(r_s\sqrt{s/n})$. Equation 29 then yields the final bound

$$\Delta_1 = \mathcal{O}(r_s\sqrt{s/n}) = \mathcal{O}\left(n^{-\beta\frac{0.78}{2}\frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}} n^{(\beta-1)/2}\right).$$

□