Local Linear Forests

Rina Friedberg*
rfriedberg@linkedin.com

Julie Tibshirani
julietibs@gmail.com

Susan Athey
athey@stanford.edu

Stefan Wager
swager@stanford.edu

September 8, 2020

Abstract

Random forests are a powerful method for non-parametric regression, but are limited in their ability to fit smooth signals. 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 valid under regularity conditions on the forest and smoothness constraints, and propose a computationally efficient construction for confidence intervals. Moving to a causal inference application, we discuss the merits of local regression adjustments for heterogeneous treatment effect estimation, and give an example on a dataset exploring the effect word choice has on attitudes to the social safety net. Last, we include simulation results on real and generated data. A software implementation is available in the R package grf.

Keywords: asymptotic normality; heterogeneous treatment effect; smoothing and nonparametric regression

*R.F. was supported by the DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. The authors also gratefully acknowledge support by the Sloan Foundation, ONR grant N00014-17-1-2131, and NSF grant DMS-1916163. S. W. was supported by a Facebook Faculty Award. The authors would like to thank Guido Imbens, Art Owen, Evan Rosenman, and Steve Yadlowsky for useful comments and discussion. R.F. is currently at LinkedIn, and this paper was included as part of her PhD dissertation at Stanford's Statistics Department.

1 Introduction

Random forests [Breiman, 2001] are a popular method for non-parametric regression that have proven effective across many application areas [Cutler et al., 2007, Díaz-Uriarte and De Andres, 2006, Svetnik et al., 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: We simulate $X_1, \ldots, X_n$ independently from the uniform distribution on $[0, 1]^{20}$, with responses

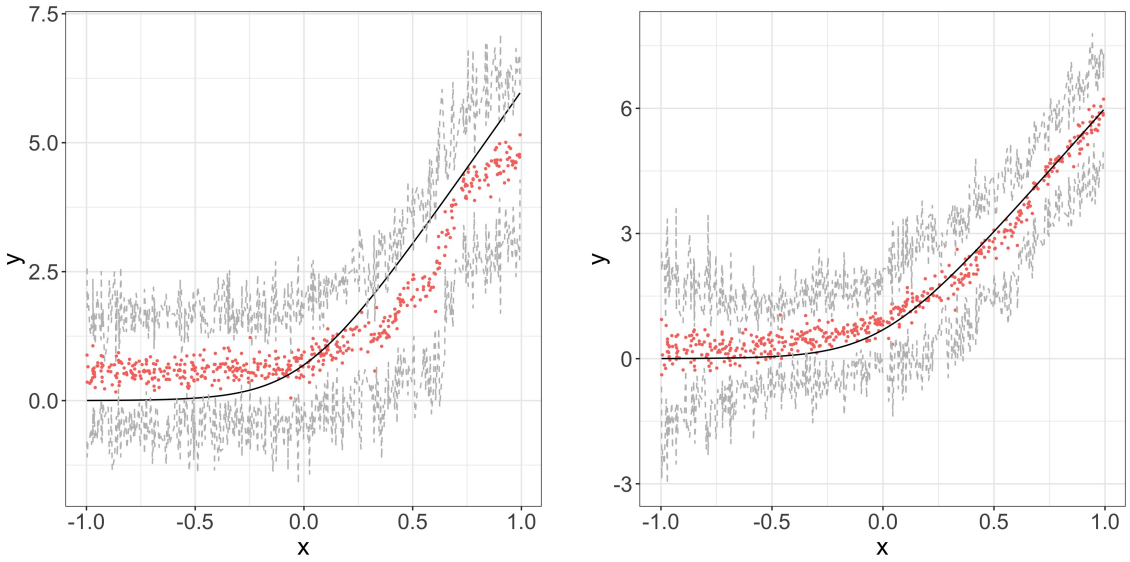
$$y_i = \log\left(1 + \exp(6X_{i1})\right) + \epsilon, \quad \epsilon \sim \mathcal{N}(0,\, 20), \tag{1}$$

and our goal is to estimate $\mu(x_0) = \mathbb{E}[Y \mid X = x_0]$. 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 Wager [2019], 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

$$\hat{\mu}_{\text{rf}}(x_0) = \sum_{i=1}^{n} \alpha_i(x_0)\, Y_i, \tag{2}$$

where the weights $\alpha_i(x_0)$, which are defined in the upcoming display (4), encode the weight given by the forest to the i-th training example when predicting at x_0. 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



Random forest Local linear forest

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 $\epsilon \sim N(0, 20)$. Forests were trained using the `R` package `grf` [Tibshirani et al., 2019] 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.

are particularly important near boundaries, where neighborhoods are asymmetric by necessity. With many covariates, these 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 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.

In their simplest form, local linear forests take the forest weights $\alpha_i(x_0)$, and use them

for local regression:

$$\begin{pmatrix} \hat{\mu}(x_0) \\ \hat{\theta}(x_0) \end{pmatrix} = \text{argmin}_{\mu,\theta} \left\{ \sum_{i=1}^{n} \alpha_i(x_0)(Y_i - \mu(x_0) - (X_i - x_0)\theta(x_0))^2 + \lambda ||\theta(x_0)||_2^2 \right\}. \quad (3)$$

Here $\hat{\mu}(x_0)$ estimates the conditional mean function $\mu(x_0)$, and $\theta(x_0)$ corrects for the local trend in $X_i - x$. The ridge penalty $\lambda||\theta(x_0)||_2^2$ 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 $\alpha_i(x_0)$, and making it account for the fact that we will use local regression to estimate $\mu(x_0)$. As a first encouraging result, in the motivating example from Figure 1, local linear forests have improved upon the fit of standard forests.

These improvements extend to many other types of forests, such as quantile regression forests [Meinshausen, 2006] or, more broadly, generalized random forests [Athey, Tibshirani, and Wager, 2019]. An extension of primary interest is to causal forests as proposed by Athey, Tibshirani, and Wager [2019], which we discuss in Section 3 in detail; other cases are analogous.

Our main formal result is a Central Limit Theorem for the predictions $\hat{\mu}(x_0)$ from a local linear forest at a given test point x, specifying the asymptotic convergence rate and its dependence on subsampling and smoothness of $\mu(x_0)$. This allows us to build pointwise Gaussian confidence intervals, giving practitioners applicable uncertainty quantification. Observe that in Figure 1, the bias of regression forest predictions affects not only the accuracy prediction curve but also the coverage corresponding confidence intervals, which are not centered on the true function. Local linear forests, in addressing this 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 those 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, they

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 6 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, 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. (This is a well-known problem. The popular core `R` function `loess` [R Core Team, 2019] allows only 1-4 predictors, while `locfit` [Loader, 2013] crashes on the simulation from (1) with $d \geq 7$.) 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 adaptivity of random forests and the ability of local linear regression to capture smoothness.

An implementation of local linear forests, compliant with the assumptions detailed in Section 4, is available in the `R` package `grf` [Tibshirani et al., 2019, R Core Team, 2019].

1.1 Empirical Example: Wage Regressions

To illustrate the promise of local linear forests, we consider the problem of predicting the logarithm of wages as a function of covariates including years of education, age, race, and gender; this function plays an important role in the study of labor markets [Heckman, Lochner, and Todd, 2003]. This problem has a mix of continuous and categorical variables, to which tree-based methods are well suited. However, wages tend to have a fairly strong and smooth association with age and education, and we might expect local regression

adjustments to help with this. In addition, the covariate space is large relative to the size of publicly available administrative data, and there are moderate to strong correlations among many of the covariates, making it challenging to obtain accurate predictions in some regions of the covariate space.

We consider data from the current population survey (CPS), available from the Minnesota Population Center [Flood et al., 2018]. These data describe the wages of 114,291 individuals in 2018 (excluding records that do not contain wage data). To evaluate how model performance varies with sample size, we divide the data into a large test set, which is used to evaluate accuracy overall and in specific regions of the covariate space, and training sets of varying sizes.

We compare local linear forests with ordinary least squares, lasso with interaction terms, gradient boosting, Bayesian additive regression trees, and random forest. For the lasso, random forests, local linear forests and boosting, we chose tuning parameters via cross-validation; in particular, for local linear forests, we tuned on leaf size and λ. Moreover, for our method, we did use a local linear correction for all variables; rather, we only used non-zero θ-coefficient in (3) for continuous predictors that had non-zero coefficients in a pilot lasso regression [Tibshirani, 1996] (in general, we have found screening of variables used for a local linear correction in (3) to benefit both the accuracy and computational performance of our approach). Table 1 compares predictive performance across several methods, showing that local linear forests can provide a predictive benefit over competing methods.

One motivation for studying wages is to compare wages across different types of workers, which requires accurate predictions even for types of workers who are less frequently observed. We thus evaluate predictive performance in several sparse regions of the covariate space, showing in Table 1 that local linear forests fit well in these regions. To further explore this idea, Figure 2 shows plots of observed log wages by predictions from ordinary least squares, lasso, random forests, and local linear forests, on individuals reporting a family size over 6, who amount for 3.3% of the observed population. Cubic spline fits for each method are included to help evaluate calibration on this relatively sparse region of the

	n_{train}	OLS	Lasso	XGB	BART	RF	LLF
	2,000	4.26 (1.43)	4.34 (0.13)	1.18 (0.06)	1.43 (0.07)	1.19 (0.07)	1.10 (0.06)
	5,000	4.21 (0.12)	4.15 (0.13)	1.17 (0.07)	1.32 (0.08)	1.15 (0.07)	1.03 (0.07)
	10,000	4.23 (0.12)	3.98 (0.11)	1.01 (0.05)	1.17 (0.07)	1.04 (0.07)	0.95 (0.06)
	50,000	4.24 (0.13)	3.98 (0.12)	0.91 (0.05)	1.05 (0.07)	0.98 (0.06)	0.92 (0.06)
	Avg. n_{test}	OLS	Lasso	XGB	BART	RF	LLF
Extreme ages	4051	1.92 (0.10)	1.74 (0.09)	0.48 (0.03)	0.52 (0.03)	0.46 (0.04)	0.44 (0.03)
Less sampled races	3547	3.90 (0.15)	3.71 (0.14)	1.02 (0.05)	1.14 (0.07)	1.01 (0.08)	0.95 (0.07)
Family size ≥ 6	894	4.55 (0.39)	4.32 (0.37)	0.85 (0.11)	1.13 (0.14)	1.06 (0.11)	0.96 (0.11)

Table 1: Mean squared error for predictions of log wages in CPS data, evaluated on a test set with 40,000 observations, for ordinary least squares (OLS), lasso with interaction terms (lasso), gradient boosting (XGB), Bayesian Additive Regression Trees (BART), random forests (RF), and local linear forests (LLF). We did a train/test split and then performed 100 replications on each method. The standard deviation across replications of the mean square error estimates is shown in parentheses. The top half of the table shows errors on training sets of size n_{train}. The lower panel shows mean square error on sparse regions of the covariate space. We fix a training dataset of size 20,000. For 100 repetitions, we draw a test set of size 40,000 and report errors on the subset of the test set corresponding to the desired condition (e.g. extreme ages). To give a relative sense of the sparsity of these regions, we report the average number of individuals in the test set.

dataset. Paired t-tests on the sets of squared errors for OLS ($t = 9.96$), lasso ($t = 8.59$), boosting ($t = 2.92$), BART ($t = 3.27$), and random forests ($t = 2.90$) give evidence for the improvements of local linear forests.

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] show how 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,

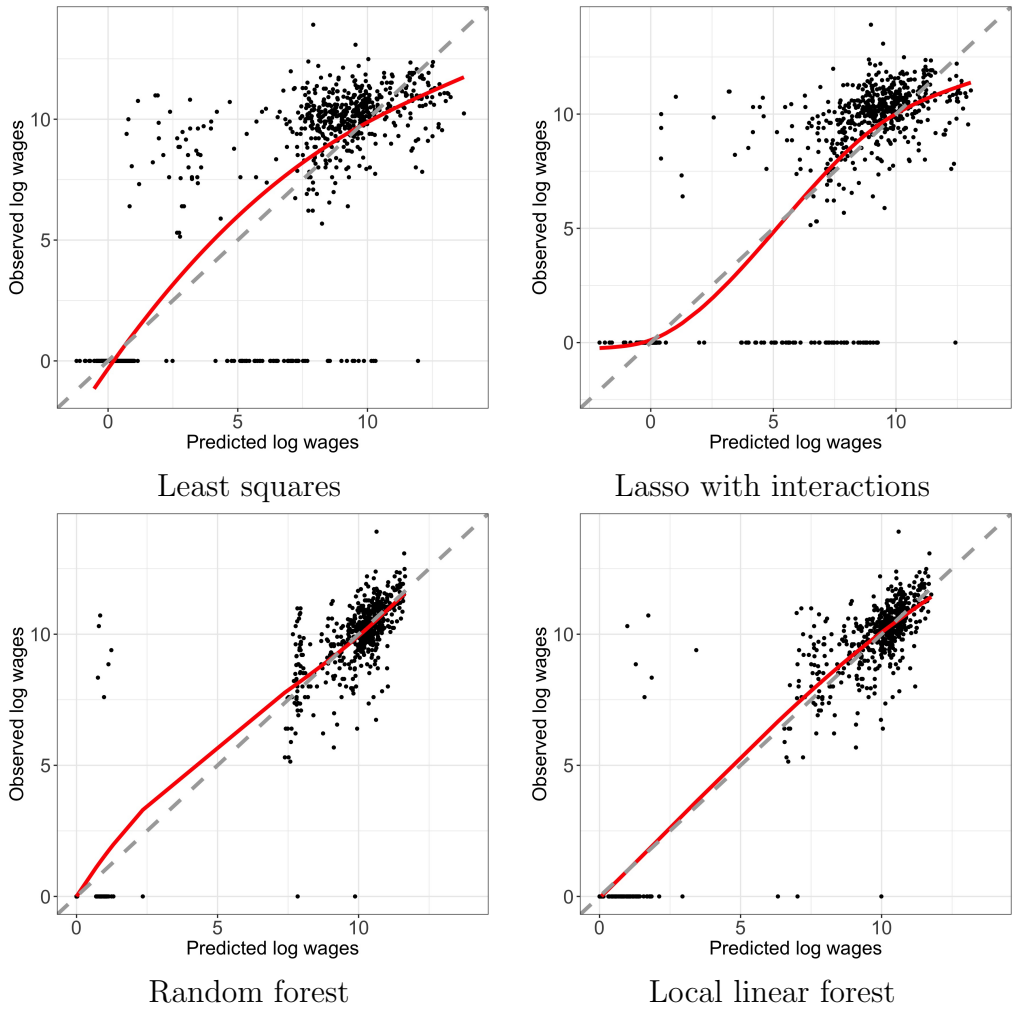


Figure 2: Observed log wages versus model predictions for test set observations, based on (left to right, top to bottom) ordinary least squares, lasso with interaction terms, random forests, and local linear forests. These algorithms were trained on 10,000 points and evaluated on all remaining test points with families over 6 people. A cubic spline fit and the diagonal 45 degree line are also shown (as the full and dashed lines, respectively); the closer the spline fit is to the diagonal, the more evidence for calibration on this test set.

and Kosorok [2015], who do deeper search during splitting to mitigate the greediness of CART. Linero and Yang [2018] propose a Bayesian regression tree ensemble tailored to learning smooth, sparse signals and prove posterior minimaxity under certain conditions, highlighting the promise of tree-based methods that can adapt to smoothness.

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 knowl-

edge no theoretical guarantees. Meinshausen [2006] proposes this technique for quantile regression forests and gives asymptotic consistency of the resulting predictions. Athey, Tibshirani, and Wager [2019] 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 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 (so that there may be asymptotic bias), while Wager and Athey [2018] provided a characterization of forests that allows for larger subsamples, deriving both asymptotic normality and valid confidence intervals. The confidence intervals proposed here are motivated by the algorithm of Sexton and Laake [2009], and build on the random forest delta method developed by Athey, Tibshirani, and Wager [2019], taking advantage of improved subsampling rates for improved coverage.

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 provide theoretical guarantees on improved 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. Menze et al. [2011] propose oblique random forests that learn split directions using the results from ridge regression, similar to our work developing splitting rules for local linear forests but more in the spirit of linear discriminant analysis (LDA). Case-specific random forests, introduced by Xu, Nettleton, and Nordman [2016], use local information to upweight training samples not at the prediction step, but during the bootstrap to generate datasets for each tree. 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 boundary 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 $Y_i = \mu(X_i) + \epsilon_i$. Consider using a random forest to estimate the conditional mean function $\mu(x_0) = \mathbb{E}[Y \mid X = x_0]$ 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 Hothorn, Lausen, Benner, and Radespiel-Troger [2004], Meinshausen [2006], and Athey, Tibshirani, and Wager [2019], entails viewing random forests as adaptive weight generators. Equivalently write $\hat{\mu}(x_0)$ as

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

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

$$\alpha_i(x_0) = \frac{1}{B} \sum_{b=1}^{B} \frac{1\{X_i \in L_b(x_0)\}}{|L_b(x_0)|} \tag{4}$$

Notice that by construction, for each i, $0 \leq \alpha_i(x_0) \leq 1$. Moreover, given that in at least one tree there exists a nonempty cell containing x_0, $\sum_{i=1}^{n} \alpha_i(x_0) = 1$; otherwise all weights are equal to zero. Athey, Tibshirani, and Wager [2019] 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 (4). Equation (3) has a closed-form solution, given below, following the closed-form solutions for ridge regression and classical local linear regression. 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 $J_{i+1,i+1} = 1$, 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

$$\begin{pmatrix} \hat{\mu}(x_0) \\ \hat{\theta}(x_0) \end{pmatrix} = \left(\Delta^T A \Delta + \lambda J\right)^{-1} \Delta^T A Y. \tag{5}$$

Define $\gamma_i = e_i \left(\Delta^T A \Delta + \lambda J\right)^{-1} \Delta^T$, where e_i is a vector of zeroes with 1 in the i-th column. Qualitatively, we can think of local linear regression as a weighting estimator, with $\gamma_i \alpha_i(x_0)$ a modulated weighting function whose x_0-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 $(x_1, Y_1), \ldots, (x_{n_P}, Y_{n_P})$. 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 them. Instead, in the parent node P, we run a ridge regression to predict Y_i from X_i:

$$\hat{Y}_i = \hat{\alpha}_P + x_i^T \hat{\beta}_P, \tag{6}$$

for intercepts $\hat{\alpha}_P$ and $\hat{\beta}_P = (x_P^T x_P + \lambda J)^{-1} x_P^T Y_P$. We then run a standard CART split on the residuals $Y_i - \hat{Y}_i$, 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 (Section 4).

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

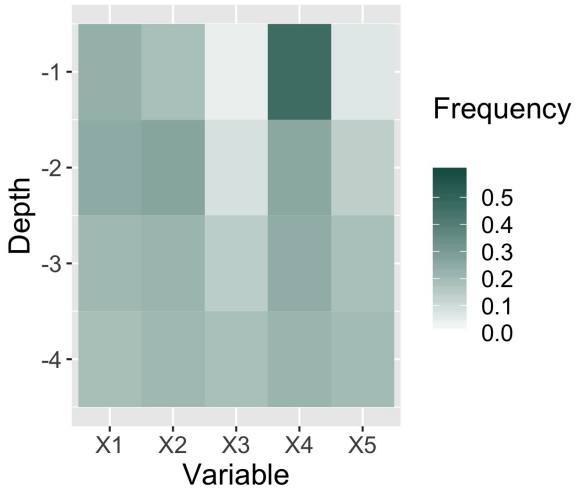
$$y = 10 \sin(\pi X_{i1} X_{i2}) + 20(X_{i3} - 0.5)^2 + 10 X_{i4} + 5 X_{i5} + \epsilon, \tag{7}$$

for $\epsilon \sim N(0, \sigma^2)$. 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 $\sin(\pi X_{i1} X_{i2})$, its ability to pick up a quadratic signal $20(X_{i3} - 0.5)^2$, and how it simultaneously models strong linear signals $10 X_{i4} + 5 X_{i5}$.

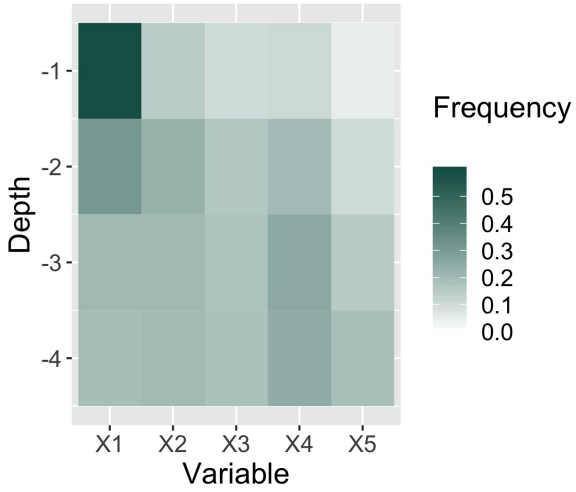
Figure 3 displays the split frequencies from an honest random forest (left) using standard CART splits, and a local linear forest (right). The x-axis is indexed by variable, here 1 through 5, and the y-axis gives tree depth for the first 4 levels of tree splits. Tiles are darkened according to how often trees in the forest split on that variable; a darker tile denotes more splits at that tree depth. CART splits very frequently on variable 4, which contributes the strongest linear signal, especially at the top of the tree but consistently throughout levels. Local linear forests rarely split on either of the strong linear signals, instead spending splits on the three that are more difficult to model.

2.2 The Value of Local Linear Splitting

Consider the following experiment, which highlights the benefit of the proposed splitting rule. We generate $X_1, \ldots, X_n$ independently and uniformly over $[0,1]^d$. We hold a cubic signal $20(X_{i1} - 0.5)^3$ constant across simulations, and on each run increase the dimension



CART split frequencies



LLF split frequencies

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 7. 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. Variables on which the forest splits frequently at depth j have a dark tile in row j.

and add another linear signal. Formally, we let $\xi_j = \mathbb{1}\{j \leq d\}$ and generate responses

$$y_i = 20(X_{i1} - 0.5)^3 \xi_1 + \sum_{j=2}^{3} 10 X_{ij} \xi_j + \sum_{j=4}^{5} 5 X_{ij} \xi_j + \sum_{j=6}^{20} 2 X_{ij} \xi_j. \tag{8}$$

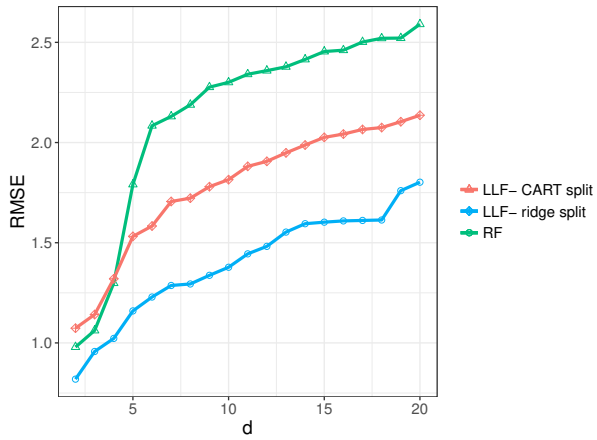


Figure 4: Results from testing different splitting rules on data generated from equation 8. Here the x-axis is dimension d, varying from 2 to 20, and we plot the Root Mean Square Error of prediction from random forests and from local linear forests with CART splits and with the ridge residual splits. We let $n = 600$ and check results on 600 test points at 50 runs for each value of d.

For example, at simulation 3 we have $\xi_1, \xi_2, \xi_3 = 1$ and hence we model $y_i = 20(X_{i1} - 0.5)^3 + 10X_{i2} + 10X_{i3}$. Root Mean Square Error is displayed in Figure 4.

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 Root Mean Square Error 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. 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.

2.3 Honest Forests

Unless noted otherwise, all random forests used in this paper are grown using a type of subsample splitting called "honesty", used by Wager and Athey [2018] to derive the asymptotic distribution of random forest prediction. As outlined in Procedure 1 of Wager and Athey

[2018], each tree in an honest forest is grown using two non-overlapping subsamples of the training data, denoted $\mathcal{I}_b$ and $\mathcal{J}_b$. We first choose a tree structure T_b using only the data in $\mathcal{J}_b$, and write $x_0 \leftrightarrow_b x'$ as the boolean indicator for whether the points x_0 and x' fall into the same leaf of T_b. Then, in a second step, we define the set of neighbors of x_0 as $L_b(x_0) = \{i \in \mathcal{I}_b : x_0 \leftrightarrow_b X_i\}$; this neighborhood function is what we then use to define the forest weights in (4). We do not use the observed outcomes y from sample $\mathcal{I}_b$ to select split points; but, to ensure that each node has a certain fraction of observations from its parent, we may use the covariates from $\mathcal{I}_b$. This modification allows us to grow honest forests that comply with the upcoming assumption 1, which says that trees are symmetric in permutations of training data index, split on every variable with nonzero probability, and balance parent observations in each child node. In this way, our theory is consistent and matches the implementation available online.

This type of subsample-splitting lets us control for potential overfitting when growing the tree T_b, because the samples $\mathcal{J}_b$ which are in the neighborhood $L_b(x_0)$ were held out when growing T_b. 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, wherein for each tree the data is split into two sets ($\mathcal{D}_n$ and $\mathcal{D}'_n$). $\mathcal{D}'_n$ is used to evaluate the CART criterion at each node during tree growth, and $\mathcal{D}_n$ is used to split. 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. All of these choices come at a cost; forests grown to smaller leaves can model meaningful signal while averaging out erroneous splits. We build honest forests by default.

Empirically, honesty can improve or worsen predictions. In particular, with small samples sizes and strong signals, honesty may limit the expressive power of forests and thus hurt predictive performance; conversely, with large sample sizes and weak signals, honesty may stabilize forests and thus improve performance (see Appendix B of Wager and Athey [2018] for a discussion). In any case, local linear corrections can help mitigate the loss of expressive power due to honesty, and so we may expect that requiring honesty is less

onerous with local linear forests than with regression forests.

2.4 Tuning a Local Linear Forest

We recommend selecting ridge penalties by cross-validation, which can be done automatically in the R package `grf`. It is often reasonable to choose different values of λ for forest training and for local linear prediction. During forest growth, equation (6) gives ridge regression predictions $x_i^T \hat{\beta}_P$ 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 (5), 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 know in advance which variables are strong predictors that should be included in the linear smoothing, 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 4 are for local linear forests trained without these heuristics.

3 Extension to Causal Forests

For conciseness, the majority of this paper focuses on local linear forests for non-parametric regression; however, 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, 2019]. To highlight this potential, we detail the method and discuss an example for heterogeneous treatment effect estimation using local linear causal forests.

As in Athey, Tibshirani, and Wager [2019], 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 $Y_i = W_i Y_i(1) + (1 - W_i) Y_i(0)$. We seek to estimate the conditional average treatment effect (CATE) of W, namely $\tau(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]$. Throughout this paper we assume uncounfoundedness [Rosenbaum and Rubin, 1983],

$$\{Y_i(0), Y_i(1)\} \perp\!\!\!\perp W_i \,|\, X_i, \tag{9}$$

and overlap, $\mathbb{P}[W_i = w] > 0$ for all i and for each $w \in \{0, 1\}$. Wager and Athey [2018] proposed an extension of random forests for estimating CATEs, and Athey, Tibshirani, and Wager [2019] 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 [2019], the orthogonalized causal forest, to strengthen its performance when $\tau(\cdot)$ is smooth.

Local linear causal forests start as orthogonalized causal forests do, by estimating the nuisance components

$$e(x_0) = \mathbb{P}[W_i = 1 \,|\, X_i = x_0] \text{ and } m(x_0) = \mathbb{E}[Y_i \,|\, X_i = x_0] \tag{10}$$

using a local linear forest. We then estimate the conditional average treatment effect via

$$\left\{\hat{\tau}(x_0),\, \hat{\theta}_\tau(x_0),\, \hat{a}(x_0),\, \hat{\theta}_a(x_0)\right\} = \text{argmin}_{\tau,\theta}\left\{ \sum_{i=1}^{n} \alpha_i(x_0)\Big(Y_i - \hat{m}^{(-i)}(X_i) - a - (X_i - x_0)\theta_a \right.$$
$$\left. - \left(\tau + \theta_\tau(X_i - x_0)\right)\left(W_i - \hat{e}^{(-i)}(X_i)\right)\Big)^2 + \lambda_\tau \left\|\theta_\tau\right\|_2^2 + \lambda_a \left\|\theta_a\right\|_2^2 \right\}, \tag{11}$$

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

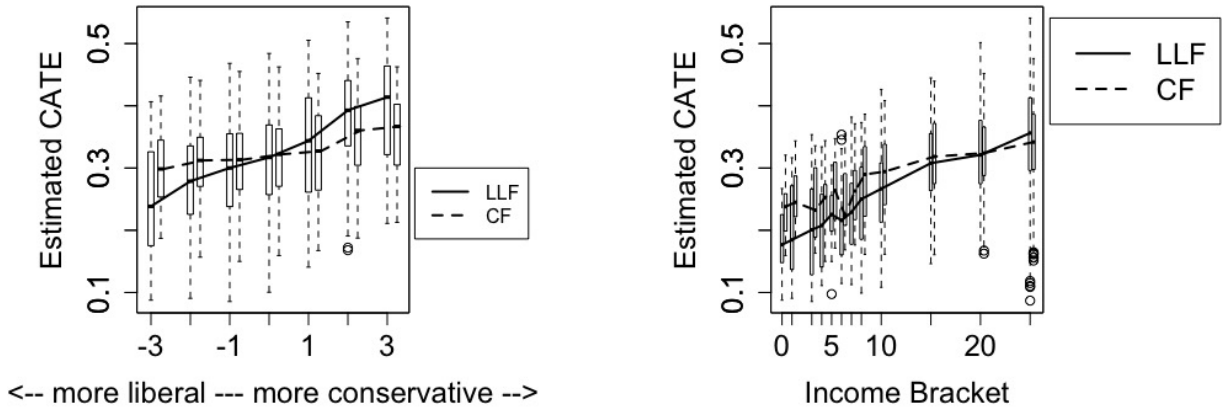
$$\widehat{\text{Err}}\left(\hat{\tau}(\cdot)\right) = \sum_{i=1}^{n}\left(Y_i - \hat{m}^{(-i)}(X_i) - \hat{\tau}(X_i)\left(W_i - \hat{e}^{(-i)}(X_i)\right)\right)^2. \tag{12}$$

Observe that, analogous to the regression case, for very large values of λ_a and λ_τ, we will recover estimates from a causal forest. From the perspective, we can see local linear causal forests as an adapted R-learner method; note that Nie and Wager [2017] and Kennedy [2020] give quasi-oracle and double-robustness properties of R-learner estimates.

3.1 Empirical Example: Attitudes to Welfare

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 [Smith et al., 2018]. Individuals filling out the survey from 1986 to 2010 answered whether they believe the government spends too much, too little, or the right amount on the social safety net. 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



CATE predictions by political views

CATE predictions by income

Figure 5: Trends in CATE predictions on the effect of the word "welfare" on people's perceptions of the social safety net. The left panel shows boxplots of CATE predictions among categories of political views, with dashed lines and straight lines connecting the medians of the causal forest and local linear causal forest predictions, respectively. The right panel shows analogous results for categories of income. All predictions are from cross-validated forests trained on 1000 training points and evaluated on 2000 test points.

$Y_i = 1$ 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, and in particular relevant for practitioners who often have more limited survey data, we test our method on smaller subsamples of the data. Figure 5 shows boxplots of CATE predictions by category of political views and income, comparing local linear causal forests and causal forests, and indicating possible heterogeneity with an approximately linear pattern.

In order to compare the performance of both methods, we use the transformed outcome metric of Athey and Imbens [2016]. Noting that $\mathbb{E}[(2W_i - 1)Y_i \mid X_i] = \tau(X_i)$, they suggest

Subsample size	200	400	800	1200	1500	2000
Causal forest	0.035	0.021	0.015	0.014	0.011	0.007
Local linear causal forest	0.027	0.017	0.013	0.013	0.011	0.006

Table 2: Estimated in-sample mean square error (13) of estimating the treatment effect on subsampled welfare data, averaged over 200 runs at each subsample size. We show estimated error from local linear causal forests and standard causal forests. Tuning parameters were selected via cross-validation using the R-learner objective.

examining the following test set error criterion

$$\begin{aligned} \mathcal{E} &= \frac{1}{|\mathcal{S}_{test}|} \sum_{i \in \mathcal{S}_{test}} \left((2W_i - 1)Y_i - \hat{\tau}(X_i)\right)^2, \\ \mathbb{E}[\mathcal{E}] &= \mathbb{E}\left[\left(\tau(X) - \hat{\tau}(X)\right)^2\right] + S_0, \quad S_0 = \mathbb{E}\left[\left((2W_i - 1)Y_i - \tau(X_i)\right)^2\right]. \end{aligned} \tag{13}$$

If we can estimate S_0 and subtract it out, then (13) 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 2 has error estimates for both types of forests using (13), and verifies that using the local linear correction improves empirical performance across different subsample sizes. Practically, we can view this change as enabling us to get good predictions on less data, a powerful improvement in cases like survey sampling where data can be expensive and difficult to attain. Section 5 contains a more detailed simulation study of local linear causal forests, comparing them with a wider array of baseline methods.

4 Asymptotic Theory

Returning to the regression case, before we delve into the main result and its proof, we briefly discuss why the asymptotic behavior of local linear forests cannot be directly derived from the existing results of Athey, Tibshirani, and Wager [2019]. 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 (5). At a first glance, $\hat{\mu}(x_0)$ indeed looks like the output of a regression forest trained on observations ρ_i. However, 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].

4.1 Main Result

We now give a Central Limit Theorem for local linear forest predictions, beginning by stating assumptions on the forest following those made in Wager and Athey [2018].

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 some probability $\pi > 0$, and the trees are grown to depth k for some $k \in \mathbb{N}$; and the trees 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 2.3, meaning that two distinct and independent subsamples are selected for each tree. Only the outcome values from one subsample are used to select the splits, and only those from the other to estimate parameters in the nodes.*

Finally, in our proof, we use the following high-level assumption on the distribution of samples as weighted by the random forest kernel. Let $d_\alpha = \sum_{i=1}^n \alpha_i(x_0)(X_i - x_0)$ denote

the difference between x_0 and the α_i-weighted average of X_i, and let $S_\alpha = \sum_{i=1}^n \alpha_i (X_i - x_0)^{\otimes 2}$ be the associated quadratic form. From Jensen's inequality, we immediately see that $d'_\alpha S_\alpha^{-1} d_\alpha \leq 1$, with equality only in the degenerate case where X_i has no variation in the direction of d_α. The following assumption rules out such degenerate distributions of X_i within leaves, and requires that the X_i enough variation along d_α to separate $d'_\alpha S_\alpha^{-1} d_\alpha$ from 1. We believe this to be a reasonable assumption that should be satisfied by any reasonable tree-growing algorithm; and it would be of interest to derive it from first principles in future work.

Assumption 3. *(Leaf Distribution) We train our forest such that* $1 - d'_\alpha S_\alpha^{-1} d_\alpha = \Omega(1)$.

Subsampling plays a central role in our asymptotic theory, allowing 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 lower bias but higher variance (and vice-versa).

In order to establish asymptotic unbiasedness of forests, Wager and Athey [2018] require a subsample size of at least n^β, with

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

This convergence rate of a traditional honest random forest, however, does not improve when $\mu(x_0)$ 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 sacrificing asymptotic variance control. This allows us to decrease the bias (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 condition $\omega \leq 0.2$ allows us to leverage theory from Wager and Athey [2018] when controlling part of $\hat{\mu}(x_0)$. We prove this result in Appendix B.

Theorem 1. *Suppose that we have training data* $Z_i = (X_i, Y_i)$ *identically and independently distributed on* $[0,1]^d \times \mathbb{R}$*, that* X_i *has a uniform distribution on* $[0,1]^d$*, and let* x_0 *be a point*

in the interior of $[0,1]^d$. *Suppose furthermore that* $\mu(x) = \mathbb{E}[Y \mid X = x]$ *is differentiable with a Lipschitz continuous derivatives* $\mu_2(x) = \mathbb{E}[Y^2 \mid X = x]$ *is Lipschitz continuous, that* $\mathrm{Var}[Y \mid X = x_0] > 0$, *and that* $\mathbb{E}[|Y - \mathbb{E}[Y \mid X = x]|^{2+\delta} \mid X = x] \leq M$ *for some constants* $M, \delta > 0$ *over all* $x \in [0,1]^d$. *Given this data-generating process, we consider local linear forests based on trees grown according to Assumptions 1, 2 and 3, with* $\omega \leq 0.2$ *and subsamples of size* s *with* $s = n^\beta$, *for*

$$\beta_{\min} = 1 - \left(1 + \frac{d}{1.3\pi} \frac{\log(\omega)}{\log(1-\omega)}\right)^{-1} < \beta < 1. \tag{16}$$

We also use a ridge regularization parameter in (3) that grows at rate

$$\lambda = \Theta\left(s^{-0.99 \frac{\log(1-\omega)}{\log(\omega)} \frac{\pi}{d}} \sqrt[4]{\frac{s}{n}}\right) \tag{17}$$

Then, there is a sequence $\sigma_n(x_0) \to 0$ *such that*

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

where $\widetilde{O}(\cdot)$ *is a version of big-O notation that ignores log-factors.*

The main draw of this results is that the best attainable rate of convergence $\beta_{\min}$ is improved compared to the rate β_{rf} from (15) obtained in Wager and Athey [2018]. The reason we were able to obtain such an improvement in the rate of convergence is that we have assumed and consequently exploited smoothness via local linear regression.

We note that the condition (16) on the subsampling rate enforces undersmoothing, i.e., that the error of $\hat{\mu}_n(x_0)$ will be dominated by variance. Undersmoothing implies that our estimator is asymptotically unbiased, and facilitates construction of confidence intervals. One limitation of this strategy is that it is in general difficult to tune an algorithm for undersmoothing; in particular, tuning via cross-validation does not guarantee undersmoothing. Developing methods for random forest inference that do not rely on undersmoothing following, e.g., Hall and Horowitz [2013], would be of considerable interest; however, this falls beyond the scope of the present paper.

4.2 Pointwise Confidence Intervals

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 [2019], 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 $\alpha_i(x_0)$:

$$\sum_{i=1}^{n} \alpha_i(x_0) \psi \left(X_i,\, Y_i;\, \hat{\mu}(x_0),\, \hat{\theta}(x_0) \right) = 0. \tag{18}$$

Athey, Tibshirani, and Wager [2019] then propose estimating the error of these estimates as

$$\widehat{\text{Var}} \left[\left(\hat{\mu}(x_0), \hat{\theta}(x_0) \right) \right] = \widehat{V}(x_0)^{-1} \widehat{H}_n(x_0) \left(\widehat{V}(x_0)^{-1} \right)', \tag{19}$$

where $V(x_0) = \nabla_{(\mu,\theta)} \mathbb{E}[\psi(x_0, Y; \mu, \theta) \mid x_0 = x_0]$ is the slope of the expected estimating equation at the optimum, and $\widehat{H}_n(x_0)$ is an estimate of

$$H_n(x_0) = \text{Var} \left[\sum_{i=1}^{n} \alpha_i(x_0) \psi \left(X_i,\, Y_i;\, \mu^*(x_0),\, \theta^*(x_0) \right) \right]. \tag{20}$$

The upshot is that $H_n(x_0)$ 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 $H_n(x_0)$ using tools originally developed for variance estimation with regression forests. Meanwhile, $V(x_0)$ can be estimated directly using standard methods.

With local linear forests, $(\hat{\mu}, \hat{\theta})$ solve (18) with score function

$$\psi(Y_i, X_i; \mu, \theta) = \nabla_{(\mu,\,\theta)} \, \frac{1}{2} \left(\left(Y_i - \Delta_i \begin{pmatrix} \mu \\ \theta \end{pmatrix} \right)^2 + \lambda \left\| \theta \right\|_2^2 \right), \tag{21}$$

where we again use notation defined in (5) and (14). First, we note that we have access to a simple and explicit estimator for $V(x_0)$: Twice differentiating (3) with respect to the

parameters $(\mu,\, \theta)$ gives

$$\nabla^2_{(\mu,\,\theta)}\ \frac{1}{2}\left(\sum_{i=1}^{n}\alpha_i(x_0)(Y_i-\mu-\Delta_i\theta)^2+\lambda||\theta||_2^2\right)=\sum_{i=1}^{n}\alpha_i(x_0)\Delta_i^T\Delta_i+\lambda J=M_\lambda, \tag{22}$$

which we can directly read off of the forest. In this paper, we are only interested in confidence intervals for $\mu(x_0)$, i.e., the first coordinate of $(\mu,\, \theta)$, and to estimate its variance we only need access to the entry in the upper-left corner of (19), which we call $\hat{\sigma}_n^2$. Given our setting, we then note that we can re-express the relevant part of (19) as follows, in terms of $\zeta' = e_1' M_\lambda^{-1}$:

$$\begin{aligned}\hat{\sigma}_n^2 &= \zeta'\widehat{H}_n(x)\zeta = \widehat{\text{Var}}\left[\sum_{i=1}^{n}\alpha_i(x_0)\ \Gamma_i(\mu^*(x_0),\, \theta^*(x_0))\right],\\ \Gamma_i(\mu,\, \theta) &= (\zeta\cdot\Delta_i)\left(Y_i-\Delta_i\begin{pmatrix}\mu\\ \theta\end{pmatrix}\right),\end{aligned} \tag{23}$$

where $\widehat{\text{Var}}[]$ refers to an estimate of the variance of the infeasible regression forest defined between the brackets.

Next, we follow Athey, Tibshirani, and Wager [2019], and proceed using the bootstrap of little bags construction of Sexton and Laake [2009] to estimate the variance of this infeasible regression forest. 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 $\mathcal{H}$:

$$\Psi_{\mathcal{H}} = \frac{1}{|\mathcal{S}_{\mathcal{H}}|}\sum_{b\in\mathcal{S}_{\mathcal{H}}}\frac{\sum_{i=1}^{n}1\left(\{X_i\in L_b(x_0)\}\right)\Gamma_i\left(\hat{\mu}(x_0),\, \hat{\theta}(x_0)\right)}{\sum_{i=1}^{n}1\left(\{X_i\in L_b(x_0)\}\right)}, \tag{24}$$

where $\mathcal{S}_{\mathcal{H}}$ is the set of trees that only use data from the half-sample $\mathcal{H}$, and $L_b(x_0)$ contains neighbors of x_0 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]

$$\hat{\sigma}_n^2 = \binom{n}{\lfloor n/2 \rfloor}^{-1} \sum_{\{\mathcal{H}:|\mathcal{H}|=\lfloor \frac{n}{2} \rfloor\}} \left(\Psi_{\mathcal{H}} - \bar{\Psi}\right)^2, \quad \bar{\Psi} = \binom{n}{\lfloor n/2 \rfloor}^{-1} \sum_{\{\mathcal{H}:|\mathcal{H}|=\lfloor \frac{n}{2} \rfloor\}} \Psi_{\mathcal{H}}. \tag{25}$$

Now, carrying out the full computation in (25) is impractical, and naive Monte Carlo approximations suffer from bias. However, as discussed in Athey, Tibshirani, and Wager [2019] 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 [2019] for details. Simulation results on empirical confidence interval performance are given in Section 5.3.

5 Simulation Study

5.1 Methods

In this section, we compare local linear forests, random forests, BART [Chipman, George, and McCulloch, 2010], and gradient boosting [Friedman, 2001]. We also include a lasso-random forest baseline for local linear forests: on half of the training data, run a lasso [Tibshirani, 1996] regression; on the second half, use a random forest to model the corresponding residuals. Like local linear forests, this method combines regression and forests, making it a natural comparison; it is similar in spirit to the tree-augmented Cox model of Su and Tsai [2005], who combine pruned CART trees with proportional hazards regression.

Random forests are trained using the R package `grf` [Tibshirani et al., 2019], and are cross-validated via the default parameter tuning in `grf`, which selects values for mtry, minimum leaf size, sample fraction, and two parameters (alpha and imbalance penalty) that control split balance. Local linear forests are tuned equivalently with additional cross-validation for regularization parameters. Variables for the regression at prediction are selected via the lasso. Because existing theoretical results for random forests rely on honesty, all random forests are built with the honest construction. All lasso models are implemented via `glmnet` [Friedman, Hastie, and Tibshirani, 2010] and cross-validated with

their automatic cross-validation feature. Local linear regression is not included in these comparisons, since the implementations `loess` and `locfit` both fail for $d > 6$ on this simulation; in Appendix A, we compare local linear regression with this set of methods on lower dimensional linear models. Unless otherwise specified, all reported errors are Root Mean Square Error on 1000 test points averaged over 50 simulation runs.

Gradient boosted trees are implemented by the R package `XGBoost` [Chen et al., 2019]. BART for treatment effect estimation is implemented following Hill [2011]. As is standard, we use the `BART` package [McCulloch et al., 2019] without any additional tuning. The motivation for not tuning is that 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.

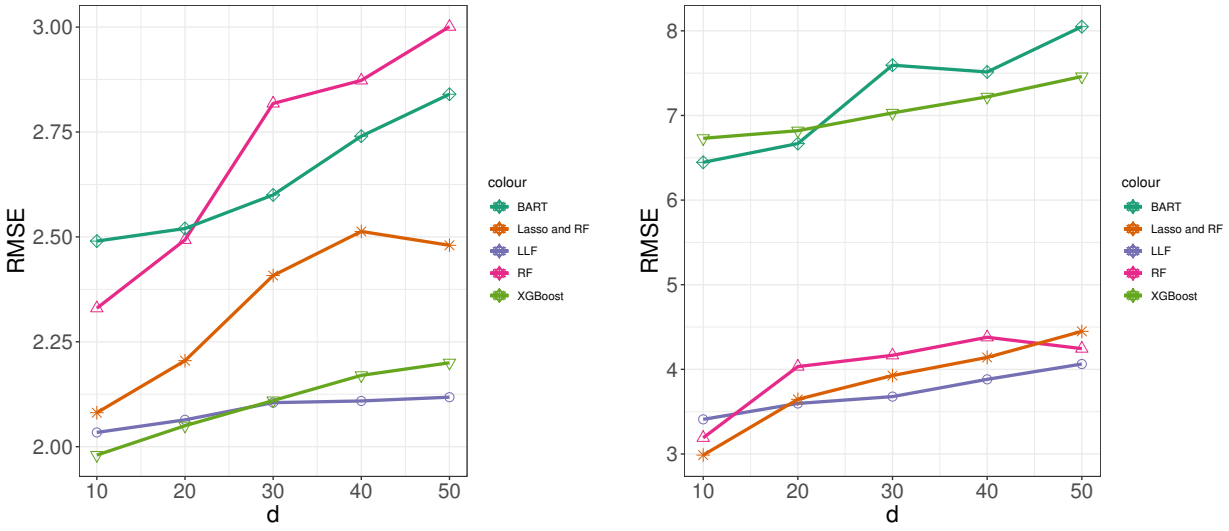
5.2 Simulation Design

The first design we study is Friedman's example from equation (7). Figure 6 shows errors at $n = 1000$ fixed, with dimension d varying from 10 to 50. There are two plots shown, to highlight the differences between error variance $\sigma = 5$ and $\sigma = 20$. Appendix A reports a grid of errors for dimensions 10, 30, and 50, with $n = 1000$ and 5000, and σ taking values of 5 and 20. The second design we consider is given in Section 1, as in equation (1). Again we test on a grid, letting dimension d take values in 5 and 50, n either 1000 or 5000, and σ at 0.1, 1, and 2. Errors are reported in Appendix A.

The third simulation 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

$$y_i = \frac{10}{1 + \exp(-10 * (X_{i1} - 0.5))} + \frac{5}{1 + \exp(-10 * (X_{i2} - 0.5))} + \epsilon, \quad \epsilon \sim N(0, 5^2). \quad (26)$$

Here we test dimension $d = 5, 20$ and values of $n = 500, 2000, 10000$. For this simulation, we compare only honest random forests and local linear forests, in order to compare confidence intervals; we compute out of bag Root Mean Square Error and average confidence interval coverage and length. Results are reported in Table 3.



Errors on low variance

Errors on high variance

Figure 6: Root Mean Square Error of predictions on 1000 test samples from equation 7, with $n = 1000$ held fixed and dimension d varied from 10 to 50. Plots is 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 random forests (RF) local linear forests (LLF), lasso and random forests, boosted trees, and BART.

5.3 Results

Figure 6 shows Root Mean Square Error from equation 7 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, lending this credibility. The lasso-random forest baseline lines up closely with local linear forests in the high noise case, separating more for low noise. BART and random forests form the next tier of methods on the low noise case; in the high noise case, honest random forest are clustered with local linear forests. BART and boosting separate in the higher noise case, suffering compared to the other methods. Appendix A shows the fuller Root Mean Square Error comparison from Friedman's model.

We move to the second simulation setup, equation 1, meant to evaluate how methods

Setup	d	n	Coverage		Length		Root Mean Square Error	
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
	5	10000	0.97	0.98	1.41	2.20	0.28	0.42
	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
	20	10000	0.97	0.99	1.23	0.89	0.24	0.13
Equation 7			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
	5	10000	0.70	0.75	2.75	2.77	1.32	1.18
	20	500	0.45	0.59	4.06	4.61	8.82	4.85
	20	2000	0.57	0.64	3.55	4.22	5.25	3.20
	20	10000	0.52	0.70	2.28	2.89	1.83	1.46
Equation 26			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
	5	10000	0.82	0.92	1.47	1.36	0.40	0.3
	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
	20	10000	0.87	0.92	1.94	1.55	0.58	0.37

Table 3: Average coverage and length of 95% confidence intervals from honest random forests (RF) and local linear forests (LLF), along with Root Mean Square Error on the same out of bag (OOB) predictions. OOB coverage is averaged over 50 runs of the simulation setups in equations 1, 7, and 26 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 7, and train on sample fraction 0.5.

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. Full error results on the range of competing methods are given in Appendix A. 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. Gradient boosted trees perform very well in the low-noise case, but their performance sharply declines when we increase σ.

We also examine the behavior of our confidence intervals in each of the given simulation setups, shown here in Table 3. We give average coverage of 95% confidence intervals from 50

repetitions of random forests and local linear forests on draws from the simulation setups in equations 1, 7, and 26, as well as average confidence interval length and Root Mean Square Error. 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 Root Mean Square Error and coverage at or above 88%. For the setup in equation 7, 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. Moreover, the poor coverage we see sometimes across both methods is likely because the confidence intervals are built on asymptotic results, which may not apply in some relatively low n settings.

We include the approximate step function in equation 26 to highlight a favorable 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 Root Mean Square Error.

The majority of these settings are well-suited to local linear forest success; one can think of several examples where the method is likely to under-perform. With a small number of covariates, the method is similar to local linear regression, possibly worse if the forest has overfit to the data. If the local linear correction does not accurately model any underlying smoothness, cross-validation will select large values for λ, but given speed and inaccurate assumptions, random forests would be preferred in this setting.

5.4 Local Linear Causal Forests

In Section 3, 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 [2019], used to evaluate their meta-learner

	Simulation 1 (equation 27)			Simulation 2 (equation 28)		
n	X-BART	CF	LLCF	X-BART	CF	LLCF
200	1.01	0.94	**0.80**	**0.67**	0.77	0.71
400	0.76	0.50	**0.47**	0.56	0.55	**0.50**
600	0.61	0.39	**0.35**	0.50	0.41	**0.38**
800	0.55	0.35	**0.31**	0.46	0.34	**0.32**
1000	0.50	0.33	**0.30**	0.44	0.29	**0.28**
1200	0.48	0.32	**0.28**	0.42	0.27	**0.26**

Table 4: Average Root Mean Square Error of predicting the heterogeneous treatment effect τ_i on 100 repetitions of the simulation given in equation (27). 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.

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

$$\tau(X_i) = \zeta(X_{i1})\zeta(X_{i2}), \quad \zeta(x) = \frac{2}{1 + \exp(-20(x - 1/3))} \tag{27}$$

$$\tau(X_i) = \zeta(X_{i1})\zeta(X_{i2}), \quad \zeta(x) = 1 + \frac{1}{1 + \exp(-20(x - 1/3))}. \tag{28}$$

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 the automatic self-tuning feature. As in the prediction simulation studies, we do not cross-validate X-BART because the authors recommend X-BART specifically for when a user does not want to carefully tune. We acknowledge that this may hinder its performance. Local linear causal forests are tuned via cross-validation. On these simulations, 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 27 as a simulation that demonstrates how forests can suffer on the boundary of a feature space, because there is a spike near $x = 0$. Root Mean Square

Error over 100 repetitions is reported in Table 4, demonstrating that 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.

6 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 as compared to generalized random forests. We showed on the welfare dataset that local linear forests can model smooth heterogeneous causal effects, and illustrated when and why they outperform competing methods. We also gave confidence intervals from the delta method for the regression 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 adjustment provides centered predictions, adjusting for errors due to an asymmetric set of neighbors. It may be that there is a useful polynomial basis corresponding to every situation where local linear forests performed well, but finding such a model would likely require 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 $\theta(x_0)$, rather than merely accounting for it to improve the precision of $\mu(x_0)$. 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. Extending our theoretical results beyond pointwise convergence would enable finding uniform confidence bands and be of considerable theoretical and practical interest.

We have also not considered the theoretical or empirical improvements that could arise from assuming higher order smoothness in the functions we are estimating; searching for additional optimality results in this setting could be another interesting research question.

References

Alberto Abadie and Guido 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 Computation*, 9(7):1545–1588, 1997.

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. *The Annals of Statistics*, 47(2):1148–1178, 2019.

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*, 115(8):1943–1948, 2018.

Gérard Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13(1):1063–1095, 2012.

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

Adam Bloniarz, Ameet Talwalkar, Bin Yu, and Christopher Wu. Supervised neighborhoods for distributed nonparametric regression. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1450–1459, Cadiz, Spain, 2016.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Leo Breiman, Jerry Friedman, Charles J. Stone, and Richard A. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, Boca Raton, FL, 1984.

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

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, and Yutian Li. *xgboost: Extreme Gradient Boosting*, 2019. URL https://CRAN.R-project.org/package=xgboost. R package version 0.82.1.

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

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

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.

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. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1982.

Bradley Efron and Charles Stein. The jackknife estimate of variance. *The Annals of Statistics*, 9(3):586–596, 1981.

Jianqing Fan and Irene Gijbels. Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 20(4):2008–2036, 1992.

Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications*. Number 66 in Monographs on statistics and applied probability series. Chapman & Hall, London, 1996.

Sarah Flood, Miriam King, Renae Rodgers, Steven Ruggles, and J. Robert Warren. Integrated public use microdata series, current population survey: Version 6.0 [dataset]. Minneapolis, MN: IPUMS, 2018. https://doi.org/10.18128/D030.V6.0, 2018.

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.

Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 03 1991.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

João Gama. Functional trees. *Machine Learning*, 55(3):219–250, 2004.

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.

Peter Hall and Joel Horowitz. A simple bootstrap method for constructing nonparametric confidence bands for functions. *The Annals of Statistics*, 41(4):1892–1921, 2013.

James Heckman, Lance Lochner, and Petra Todd. Fifty years of mincer earnings regressions. NBER Working Papers 9732, National Bureau of Economic Research, Inc, 2003.

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*, 58(301):13–30, 1963.

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

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

Aram Karalič. Employing linear regression in regression tree leaves. In *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 440–441, New York, NY, USA, 1992. John Wiley & Sons, Inc.

Edward H. Kennedy. Optimal doubly robust estimation of heterogeneous causal effects. *arXiv:2004.14497*, 2020.

Soren R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Meta-learners for Estimating Heterogeneous Treatment Effects using Machine Learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, 2019.

Yehua Li and Tailen Hsing. Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *The Annals of Statistics*, 38(6):3321–3351, 2010.

Antonio Ricardo Linero and Yun Yang. Bayesian Regression Tree Ensembles that Adapt to Smoothness and Sparsity. *Journal of the Royal Statistical Society, Series B*, 80(5): 1087–1110, 2018.

Catherine Loader. *locfit: Local Regression, Likelihood and Density Estimation.*, 2013. R package version 1.5-9.1.

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

Robert McCulloch, Rodney Sparapani, Robert Gramacy, Charles Spanbauer, and Matthew Pratola. *BART: Bayesian Additive Regression Trees*, 2019. R package version 2.4.

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

Lucas Mentch and Giles Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17(1):841–881, 2016.

Bjoern H. Menze, B. Michael Kelm, Daniel N. Splitthoff, Ullrich Koethe, and Fred A. Hamprecht. On oblique random forests. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 453–469, Berlin, Heidelberg, 2011. Springer.

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

Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *arXiv preprint arXiv:1712.04912*, 2017.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

Peter M Robinson. Root-n-consistent semiparametric regression. *Econometrica*, 56(4): 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.

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.

Erwan Scornet, Gerard Biau, and Jean-Philippe Vert. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015.

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

Tom W. Smith, Michael Davern, Jeremy Freese, and Michael Hout. General social surveys, 1972-2016 [machine-readable data file]. /Principal Investigator, Smith, Tom W.; Co-Principal Investigators, Peter V. Marsden and Michael Hout; Sponsored by National Science Foundation. –NORC ed.– Chicago: NORC: NORC at the University of Chicago [producer and distributor]. Data accessed from the GSS Data Explorer website at gss-dataexplorer.norc.org, 2018.

Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4): 595–620, 1977.

Xiaogang Su and Chih-Ling Tsai. Tree-augmented Cox proportional hazards models. *Biostatistics*, 6(3):486–499, 2005.

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.

Matt Taddy, Chun-Sheng Chen, Jun Yu, and Mitch Wyle. Bayesian and empirical Bayesian forests. *Proceedings of the 32nd International Conference on Machine Learning*, pages 967–976, 2015.

Julie Tibshirani, Susan Athey, Rina Friedberg, Vitor Hadad, Luke Miner, Stefan Wager, and Marvin Wright. *grf: Generalized Random Forests (Beta)*, 2019. R package version 0.10.3.

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.

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:1503.06388*, 2015.

Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*, 15 (1):1625–1651, 2014.

Ruo Xu, Dan Nettleton, and Daniel J. Nordman. Case-specific random forests. *Journal of Computational and Graphical Statistics*, 25(1):49–65, 2016.

Fang Yao, Hans-Georg Muller, and Jane-Ling Wang. Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33(6):2873–2903, 2005.

Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, 2008.

Yichen Zhou and Giles Hooker. Boulevard: Regularized stochastic gradient boosted trees and their limiting distribution. *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, boosting, random forests, and local linear forests, on Friedman's data-generating process: generate $X_1, \ldots, X_n$ i.i.d. $U[0,1]^5$ and model Y_i from

$$y = 10 \sin(\pi X_{i1} X_{i2}) + 20(X_{i3} - 0.5)^2 + 10 X_{i4} + 5 X_{i5} + \epsilon,$$

Errors are reported on dimension ranging from 10 to 50, σ from 5 to 20, and $n = 1000$ and 5000, averaged over 50 training runs.

d	n	σ	RF	lasso-RF	LLF	BART	XGBoost
10	1000	5	2.33	2.12	2.03	2.49	**1.98**
10	5000	5	1.90	**1.48**	1.57	1.51	1.52
30	1000	5	2.82	2.41	**2.11**	2.60	**2.11**
30	5000	5	2.08	**1.61**	1.73	2.03	1.64
50	1000	5	3.00	2.48	**2.12**	2.84	2.20
50	5000	5	2.18	1.82	**1.80**	2.11	1.82
10	1000	20	**3.19**	3.41	3.40	6.45	6.73
10	5000	20	2.43	2.35	**2.29**	3.85	4.42
30	1000	20	4.17	3.98	**3.68**	7.60	7.03
30	5000	20	2.97	2.66	**2.40**	4.78	4.85
50	1000	20	4.25	4.45	**3.88**	8.05	7.47
50	5000	20	3.16	2.67	**2.35**	4.95	4.97

Table 5: Root mean square error on Friedman's function, with dimension d 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. 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 $n = 1000$ and $n = 5000$ 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 the main document. 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, boosting, random forests, and local linear forests, on the

data-generating process: simulate $X_1, \ldots, X_n$ i.i.d. Uniform $[0,1]^{20}$, with responses

$$y_i = \log\left(1 + \exp(6X_{i1})\right) + \epsilon, \quad \epsilon \sim \mathcal{N}(0,\, 20).$$

Errors are reported on dimension ranging from 5 to 20, σ from 0.1 to 2, and $n = 1000$ and 5000, averaged over 50 training runs.

d	n	σ	RF	lasso- RF	LLF	BART	XGBoost
5	1000	0.1	0.10	0.06	**0.02**	0.27	0.07
5	5000	0.1	0.06	**0.02**	**0.02**	0.22	0.06
50	1000	0.1	0.29	0.18	0.11	0.52	**0.07**
50	5000	0.1	0.18	0.10	0.07	0.62	**0.06**
5	1000	1	0.21	0.24	**0.14**	0.47	0.56
5	5000	1	0.15	0.11	**0.09**	0.26	0.52
50	1000	1	0.41	0.39	**0.20**	0.82	0.53
50	5000	1	0.23	0.21	**0.10**	0.57	0.52
5	1000	2	0.31	0.55	**0.26**	0.69	1.21
5	5000	2	0.25	0.28	**0.21**	0.40	1.18
50	1000	2	0.47	0.27	**0.24**	0.89	1.22
50	5000	2	0.33	0.27	**0.15**	0.70	0.96

Table 6: Root mean square error from simulations on random forests, lasso-random forest, local linear forests, BART, and boosting. We vary sample size n, error variance σ, and ambient dimension d, 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.

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

$$y_i = 10X_{i1} + 5X_{i12} + X_{i3} + \epsilon \tag{29}$$

$$y_i = 10X_{i1} + 5X_{i2}^2 + X_{i3}^3 + \epsilon, \tag{30}$$

where $\epsilon \sim N(0, \sigma^2)$ and $\sigma \in \{1, 5, 10\}$. Root mean square error 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 29, we see

that lasso outperforms the other methods, as we would expect; in the polynomial given in equation 30, 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 29	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 30	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: Root Mean Square Error from simulations on equations 29 and 30 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.

B Proof of Theorem 1

Throughout this proof, we use the notation M_λ established in (19), and shorthand $Y_i = \mu(X_i) + \epsilon_i$. 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 around $\mu(x)$ around x_0 yields the following decomposition starting from (5):

$$\begin{aligned}
\hat{\mu}(x_0) &= e_1^T M_\lambda^{-1} \sum_{i=1}^n \begin{pmatrix} 1 \\ X_i - x_0 \end{pmatrix} \alpha_i(x_0) Y_i = \mu(x_0) + \hat{\gamma}_n(x_0) + Q(x_0) + O\left(\bar{R}^2\right), \\
\hat{\gamma}_n(x_0) &= e_1^T M_\lambda^{-1} \sum_{i=1}^n \begin{pmatrix} 1 \\ X_i - x_0 \end{pmatrix} \alpha_i(x_0) \epsilon_i, \\
Q(x_0) &= e_1^T M_\lambda^{-1} \sum_{i=1}^n \begin{pmatrix} 1 \\ X_i - x_0 \end{pmatrix} \alpha_i(x_0) \left(\nabla \mu(x_0) \cdot (X_i - x_0)\right),
\end{aligned} \tag{31}$$

where $\bar{R}^2$ is the average squared radius of leaves T_b in the forest. In other words, we have decomposed our forest into a variance term $\hat{\gamma}_n(x_0)$, a regularization bias term $Q(x_0)$, and

a curvature bias term that's bounded on the order of $\bar{R}^2$. Our main goal is to show that we can approximate $\hat{\gamma}_n(x_0)$ via an (infeasible) regression forest, while the remaining terms are lower order. For simplicity, moving forward we will write $\alpha_i(x_0) = \alpha_i$, dropping the written dependence on x_0.

Curvature bias To control the curvature bias, we need to control the radius R_{T_b} of a typical leaf containing x_0. To do so, we use the following bound. Recall that $X_1, \ldots, X_s \sim U([0,1]^d)$ independently, and that T_b is a regular, random-split tree. By Lemma 2 of Wager and Athey [2018], we then see that for any $0 < \eta < 1$ and for large enough s,

$$\mathbb{P}\left[\text{diam}_j(L(x_0)) \geq \left(\frac{s}{2k-1}\right)^{-\frac{0.99(1-\eta)\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}}\right] \leq \left(\frac{s}{2k-1}\right)^{-\frac{\eta^2}{2}\frac{1}{\log(\omega^{-1})}\frac{\pi}{d}}, \tag{32}$$

where k is (fixed) the tree-depth parameter from Assumption 1. We start by applying (32) with $\eta = 0.49$, and note that $0.99(1-49) > 0.5$ and $0.49^2/2/\log(1/0.8) > 0.53$, meaning that for all $\omega \leq 0.2$,

$$\mathbb{P}\left(\text{diam}_j(L(x_0)) \geq r_s\right) \leq r_s^{1.06}, \qquad r_s = s^{-\frac{1}{2}\frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})}\frac{\pi}{d}}. \tag{33}$$

This suggests that most leaves should have radius bounded on the order of r_s. To get a useful bound on the second moment of leaf radii via $\bar{R}^2$, though, we need to use chaining: Setting $\eta = 0.71$, we find that

$$\mathbb{P}\left(\text{diam}_j(L(x_0)) \geq r_s^{0.57}\right) \leq r_s^{2.2}.$$

Then, applying Markov's inequality twice, we see that $\bar{R}^2 = O_p(r_s^2)$.

Regularization bias The term $Q(x_0)$ in (31) has more intricate behavior. We note that, if we had no regularization at all, then the local linear correction would perfectly adjust for the slope of $\mu(\cdot)$ and x_0, and so we would have $Q(x_0) = 0$; unfortunately, however, we need positive regularization in other parts of the proof so we cannot directly use this

fact. Conversely, as $\lambda \to \infty$, the local linear forest becomes a regression forest, and $Q(x_0)$ becomes a bias term on the order of $\bar{R}$; and this was the dominant bias term in the analysis of Wager and Athey [2018].

The derivation shows that, given a reasonable amount of regularization $0 < \lambda < \infty$, the term $Q(x_0)$ is non-zero but still much smaller than $\bar{R}$. Recall our notation Δ_i denoting a $p+1$-dimensional vector consisting of a 1 stacked with $X_i - x_0$, and let $v = (0, \nabla\mu(x_0))$. Then, writing Δ for the matrix with rows Δ_i and plugging in the expression 19 for M_λ, we see that

$$\begin{aligned}
Q(x_0) &= e_1' \left(\Delta' A \Delta + \lambda J\right)^{-1} \Delta' A \Delta v \\
&= -e_1' \left(\Delta' A \Delta + \lambda J\right)^{-1} \lambda J v \\
&= -\lambda e_1' \left(\Delta' A \Delta + \lambda J\right)^{-1} v \\
&= \lambda \left(1 - d_\alpha' \left(S_\alpha + \lambda I\right)^{-1} d_\alpha\right)^{-1} d_\alpha' \left(S_\alpha + \lambda I\right)^{-1} \nabla\mu(x_0),
\end{aligned}$$

where the last line followed from the Schur formula, with notation $d_\alpha = \sum_{i=1}^n \alpha_i (X_i - x_0)$ and $S_\alpha = \sum_{i=1}^n \alpha_i (X_i - x_0)^{\otimes 2}$ as used in Assumption 3. We now make some observations. First, by Assumption 3

$$\left(1 - d_\alpha' \left(S_\alpha + \lambda I\right)^{-1} d_\alpha\right)^{-1} = O_p(1)$$

is of constant order in probability. Second, by Cauchy-Schwarz,

$$\begin{aligned}
d_\alpha' \left(S_\alpha + \lambda I\right)^{-1} \nabla\mu(x_0) &\leq \sqrt{d_\alpha \left(S_\alpha + \lambda I\right)^{-1} d_\alpha} \sqrt{\nabla\mu(x_0)' \left(S_\alpha + \lambda I\right)^{-1} \nabla\mu(x_0)} \\
&\leq \lambda^{-1/2} \left\|\nabla\mu(x_0)\right\|_2,
\end{aligned}$$

noting that $d_\alpha' S_\alpha^{-1} d_\alpha \leq 1$ by Jensen's inequality. Combining all these facts together, we find that $Q(x_0) = O_p(\sqrt{\lambda})$.

The variance term Finally, we turn to the variance term $\hat{\gamma}_n(x_0)$. To do so, our main task is to couple $\hat{\gamma}_n$ with an approximation $\tilde{\gamma}_n$, defined as

$$\tilde{\gamma}_n(x_0) = \sum_{i=1}^{n} \alpha_i \tilde{Y}_i, \quad \text{where} \quad \tilde{Y}_i = e_1^T \mathbb{E}[M_\lambda]^{-1} \begin{pmatrix} 1 \\ X_i - x_0 \end{pmatrix} \epsilon_i. \tag{34}$$

Now, we note that $\tilde{Y}_i$ is independent of α_i conditionally on X_i (because the problematic associations discussed at the beginning of Section 4 were mediated by M_λ), and so $\tilde{\gamma}_n(x_0)$ is just the prediction made by a "regression forest" with outcome $\tilde{Y}_i$. Consequently $\tilde{\gamma}_n$ can be characterized via standard tools used to study random forests.

We sketch out an argument below, based on the fact that M_λ concentrates around its expectation. Following the line of argumentation in Wager and Athey [2018], we see that M_λ is a U-statistic with kernel size s. Moreover, by (33), we see that the stochastic fluctuations of the terms forming M_λ are of order r_s^2. Thus, we can use concentration inequalities for U-statistics following Hoeffding [1963] to verify that (to use this concentration inequality, we need to perform several steps of chaining following (33), going up to $\eta = 0.98$)

$$\left\| M_\lambda - \mathbb{E}[M_\lambda] \right\|_\infty = O_p\left(r_s^2 \sqrt{s/n} \right). \tag{35}$$

Next, note that

$$\hat{\gamma}_n(x_0) - \tilde{\gamma}_n(x_0) = e_1 \left(M_\lambda^{-1} - \mathbb{E}[M_\lambda]^{-1} \right) \Delta' A \epsilon. \tag{36}$$

Thus, because ϵ is independent of all other terms in (36), we see that the discrepancy between $\hat{\gamma}_n(x_0)$ and $\tilde{\gamma}_n(x_0)$ is bounded on the order of $\left\| e_1 \left(M_\lambda^{-1} - \mathbb{E}[M_\lambda]^{-1} \right) \Delta' A \right\|_2$; an application of the Schur formula together with (35) then implies that

$$\hat{\gamma}_n(x_0) - \tilde{\gamma}_n(x_0) = O_p\left(\lambda^{-2} r_s^4 \, s/n \right) \tag{37}$$

for all $\lambda \gg r_s^2 \sqrt{s/n}$.

Wrapping up We are now ready to put everything together. Given everything we've seen so far, we've established that

$$\hat{\mu}(x_0) - \mu(x_0) = \tilde{\gamma}(x_0) + O_p\left(r_s^2 + \sqrt{\lambda} + \lambda^{-2} r_s^4 \frac{s}{n}\right)$$

for all $\lambda \gg r_s^2 \sqrt{s/n}$. Thus, setting $\lambda = \Theta(r_s^{1.98} \sqrt[4]{s/n})$ as in (17), we get

$$\hat{\mu}(x_0) - \mu(x_0) = \tilde{\gamma}(x_0) + O_p\left(r_s^2 + r_s^{0.99} \sqrt[8]{s/n} + r_s^{0.04} \sqrt{s/n}\right).$$

Now, recall that we have chose $s = n^\beta$ for some $\beta \geq \beta_{\min}$, meaning that

$$\sqrt[3/8]{s/n} = s^{\frac{3(1-\beta^{-1})}{8}} \geq s^{-\frac{3\times 1.3}{8} \frac{\log((1-\omega)^{-1})}{\log(\omega^{-1})} \frac{\pi}{d}} \gg r_s^{0.99},$$

and so the above expression simplifies to

$$\hat{\mu}(x_0) - \mu(x_0) = \tilde{\gamma}(x_0) + o_p\left(\sqrt{s/n}\right). \tag{38}$$

It remains to show that $\tilde{\gamma}(x_0)$ is asymptotically centered and Gaussian with errors on the scale of $\sqrt{s/n}$, meaning that $\tilde{\gamma}(x_0)$ is in fact the dominant error term in $\hat{\mu}(x_0)$.

But now, recall that $\tilde{\gamma}(x_0)$ is simply a regression forest with outcome $\tilde{Y}_i$. Thus, Theorem 8 of Wager and Athey [2018] directly implies that there is sequence $\sigma_n(x_0) \to 0$ such that

$$\frac{\tilde{\gamma}_n(x_0)}{\sigma_n(x_0)} \Rightarrow \mathcal{N}(0, 1); \tag{39}$$

here, we used the fact that the ϵ_i are all mean-zero conditionally on the tree construction, and so $\mathbb{E}[\tilde{\gamma}(x_0)] = 0$. Finally, from Theorem 5 of Wager and Athey [2018], we see that $\sigma_n(x_0) = \sqrt{s/n}\,\text{polylog}(s)$, and we note that our above argument in fact established a polynomial gap between the error term in (38) and $\sqrt{s/n}$. Thus (39) in fact captures the dominant error term of our estimator.