University of North Carolina at Chapel Hill
The University of North Carolina at Chapel Hill Department of Biostatistics Technical Report Series

Year 2012 *Paper* 33

Reinforcement Learning Trees

Ruoqing Zhu* Donglin Zeng†
Michael R. Kosorok‡

*University of North Carolina at Chapel Hill, rzhu@live.unc.edu
†University of North Carolina at Chapel Hill, dzeng@bios.unc.edu
‡University of North Carolina at Chapel Hill, kosorok@unc.edu

Reinforcement Learning Trees

Ruoqing Zhu, Donglin Zeng, and Michael R. Kosorok

Abstract

In this paper, we introduce a new type of tree-based regression method, reinforcement learning trees (RLT), which exhibits significantly improved performance over traditional methods such as random forests (Breiman, 2001). The innovations are three-fold. First, the new method implements reinforcement learning at each selection of a splitting variable during the tree construction processes. By splitting on the variable that brings the greatest future improvement in later splits, rather than choosing the one with largest marginal effect from the immediate split, the constructed tree utilizes the available samples in a more efficient way. Moreover, such an approach can be adapted to make high-dimensional cuts available at a relatively small computational cost. Second, we propose a variable screening method that progressively mutes noise variables during the construction of each individual tree. The muting procedure also takes advantage of reinforcement learning and prevents noise variables from being considered in the search for splitting rules, so that towards a terminal node when the sample size is small, the splitting rules are still constructed from only strong variables. Last, we investigate asymptotic properties of the proposed method. We can show that under the proposed splitting variable selection procedure, the constructed trees are consistent. The error bounds for the proposed RLT are shown to depend on a pre-selected number p_0, where p_0 is an educated guess of the number of strong variables which is usually much smaller than the total number of variables p but at least as large as the true number of strong variables p_1. Hence when p_0 is properly chosen, the error bounds can be significantly improved.

Submitted to the Annals of Statistics

REINFORCEMENT LEARNING TREES

By Ruoqing Zhu, Donglin Zeng and Michael R. Kosorok

University of North Carolina at Chapel Hill

In this paper, we introduce a new type of tree-based regression method, reinforcement learning trees (RLT), which exhibits significantly improved performance over traditional methods such as random forests (Breiman, 2001). The innovations are three-fold. First, the new method implements reinforcement learning at each selection of a splitting variable during the tree construction processes. By splitting on the variable that brings the greatest future improvement in later splits, rather than choosing the one with largest marginal effect from the immediate split, the constructed tree utilizes the available samples in a more efficient way. Moreover, such an approach can be adapted to make high-dimensional cuts available at a relatively small computational cost. Second, we propose a variable screening method that progressively mutes noise variables during the construction of each individual tree. The muting procedure also takes advantage of reinforcement learning and prevents noise variables from being considered in the search for splitting rules, so that towards a terminal node when the sample size is small, the splitting rules are still constructed from only strong variables. Last, we investigate asymptotic properties of the proposed method. We can show that under the proposed splitting variable selection procedure, the constructed trees are consistent. The error bounds for the proposed RLT are shown to depend on a pre-selected number p_0, where p_0 is an educated guess of the number of strong variables which is usually much smaller than the total number of variables p but at least as large as the true number of strong variables p_1. Hence when p_0 is properly chosen, the error bounds can be significantly improved.

1. Introduction. In high-dimensional settings, the concept of sparsity, that there is a relatively small set of variables which completely convey the true signal, is both intuitive and useful. Many variable selection methods have been proposed to identify this set of true signal variables. Among these methods, tree-based approaches have drawn much attention in the literature due to their capacity for handling sparsity without too much overfitting. However, when the high-dimensional signal surface is arbitrarily complicated, traditional tree-based method can either fail to detect the true signals or may use the data in an inefficient way. This is caused by two fundamental d-

Keywords and phrases: Reinforcement Learning, Tree Methods, Random Forests, Consistency, Error Bound

ifficulties. First, at each split of the tree, only marginal effects are considered. This can potentially result in overlooking complicated interactions. Second, towards terminal nodes, as the sample size decreases dramatically, it is nearly impossible to identify the strong variables in such a high-dimensional space. As a result, the search for best splitting rules can perform as badly as random selection. To avoid the above mentioned drawbacks, we attempt in this paper to embed reinforcement learning (Sutton and Barto, 1998) into random forests (Breiman, 2001) to obtain significant reduction in prediction error. Before outlining the proposed method, we briefly review previous work that prepares the way.

Over the last few decades, tree-based methods have seen a remarkable revolution. Ensemble methods (Breiman, 1996) were introduced to improve the original classification and regression trees (CART) proposed by Breiman et al. (1984). Later, a series of works including (Amit and Geman, 1997; Breiman, 2000; Dietterich, 2000) paved the way for the introduction of random forests (Breiman, 2001), a state-of-the-art ensemble method. In this approach, multiple trees are built based on independently generated bootstrap samples. When building each tree at each internal node, a set of variables is randomly selected and the best cut point in this selected set is chosen to create a splitting rule which generates two subsequent daughter nodes. Each tree is grown to full size with a pre-specified minimal terminal node sample size as a stopping rule. All trees are averaged to acquire a final prediction.

Random forests have garnered significant popularity due to their accuracy and capacity to handle high dimensional data. Many versions of random forests have been proposed, such as perfect random forests by Cutler and Zhao (2001), which have exactly one observation in each terminal node; Extremely randomized trees (ERT) by Geurts et al. (2006), which use random cut points rather than searching for the best cut point; and Bayesian additive regression trees (BART) by Chipman et al. (2010), which integrates tree-based methods into a Bayesian framework. Experiments on a variety of methods have led to a general belief that an accurate tree-based method has a good blend of greediness (data adaptivity) and diversity (randomness).

While methodologies for tree-based methods have been actively studied, the asymptotic behavior of random forests has also started to draw significant interest. Lin and Jeon (2006) established the connection between random forests and nearest neighborhood estimation. They also established a lower bound on the convergence rate of random forests under a special type of tree construction mechanism. Biau et al. (2008) proved consistency for a variety of types of random forests, including purely random forests (PRF). However, this author also provide an example which demonstrates inconsis-

tency of trees under certain greedy construction rules. One important fact to point out is that the consistency and convergence rate of random forests (including but not limited to the original version proposed by Breiman (2001)) heavily rely on the particularly implemented splitting rule. For example, purely random forests, where splitting rules are random and independent from training samples, provide a much more friendly framework for analysis compared to the original random forests. The cost of this complete randomness is inefficiency in detecting the true model structure because most of the splits are likely to select noise variables when the underlying model structure is sparse. As the splitting rules become greedy, the splitting path from a root node to a terminal node becomes extremely complicated and depends on the exact true model structure. This is also the reason why the asymptotic properties of the original random forests remain unclear. Up to now, there appears to be no tree-based method possessing both established theoretical validity and excellent practical performance.

Following the discussions by Lin and Jeon (2006) and Breiman (2004) on a special type of purely random forest, Biau (2012) proved consistency and showed that the convergence rate only depends on the number of strong variables which, collectively, completely define the true model structure. The proof for the convergence rate result in his paper can serve as a guideline for future analysis of random forests under more general structures. However, behind this celebrated result, two key components require careful further investigation. First, the probability of using a strong variable to split at an internal node depends on the within-node data (which possibly depends on an independent sample as suggested in Biau (2012)). With rapidly reduced sample sizes towards terminal nodes, this probability is unlikely to behave well for the entire tree. However, a large terminal node size is likely to introduce increased bias which may also harm the error rate. Second, identifying strong variables in a high dimensional surface can still be very tricky. The counterexample of consistency given by Biau et al. (2008) can penitentially lead to blinding of the selection criteria so that strong variables may not be chosen. As we explained at the beginning of this article, the rationale behind the above argument is that one cannot fully explore a high dimensional surface from a viewpoint which only assesses the marginal effect of each variable. Hence if the marginal effect of a strong variable is behaving like a noise variable, then the selection process may fail.

Assuming that we have p variables, among which, there are p_1 strong variables and p_2 noise variables, it is easy to see that a good single tree only splits on the p_1 strong variables, since any cut on the noise variables is a waste of sample size and will likely increase the error rate. Intuitively,

the tree should be constructed in a way that the variables carrying stronger signals have more cuts. Later, we will need to give a formal definition of "signal", which essentially means that the variable contributes to the structure of the true prediction function. For example, assume that we have a linear model with the expected outcome variable $E(Y) = \beta_0 + \mathbf{X}^T\boldsymbol{\beta}$. Note that in this model, β's should represent the signal strength (variable importance), and the terminal node interval length for each variable should be negatively proportional to the magnitude of the corresponding β's (Lin and Jeon, 2006), which means that the variables associated with larger β's should receive more cuts. Considering that a typical tree only cuts on one variable at each internal node, the question becomes: can we always identify the most important variable in the current node?

In this paper, we introduce a new philosophy—reinforcement learning—into the tree-based model framework. For a comprehensive review of reinforcement learning within the artificial intelligence field in computer science and statistical learning, we refer to Sutton and Barto (1998). An important characteristic of reinforcement learning is the "peek-at-the-future" notion which benefits the long-term performance rather than short-term performance. The main features we will employ in the proposed method are: first, to choose variable(s) for each split which will bring the largest return from future branching splits rather than only focusing on the immediate consequences of the split. Such a splitting mechanism can break any hidden structure and avoid inconsistency by forcing splits on strong variables even if they do not show any marginal effect; second, progressively muting noise variables as we go deeper down a tree so that even as the sample size decreases rapidly towards a terminal node, the strong variable(s) can still be properly identified from the reduced space. One consequence of the new approach, which we call reinforcement learning trees (RLT), as we will show later, is that the convergence rate should not depend on p, but instead, it depends on a pre-specified value p_0 which is much smaller than p and larger than p_1. Hence, when p_0 is properly chosen, the convergence rate can be greatly improved.

Another extension we bring with the proposed RLT is a high-dimensional cut which uses a linear combination of variables to create a splitting rule. In traditional tree-based methods, searching for a high-dimensional cut will dramatically increase the computational intensity. However, with the pre-identification of important variables, the cutting surface can be reasonably formed without exhaustive searching. In the simulation studies and data analyses presented later, we will examine the performance of the newly proposed RLT with both one-dimensional and high-dimensional cuts and show

that the benefit can be profound in some situations.

The paper is organized as follows. In Section 2, we introduce the underlying model and notation to facilitate the formulation of our method. In Section 3, we give details of the methodology for the proposed approach. Theoretical results and their interpretation are given in Section 4. Most of the details of the proofs will be deferred to the last section. In Sections 5 we compare RLT with popular statistical learning tools, such as random forests (Breiman, 2001), BART (Chipman et al., 2010), gradient boosting (Friedman, 2001) and GLM with LASSO (Efron et al., 2004), using simulation studies and real datasets. Section 6 contains some discussion and gives rationale for both the method and asymptotic behaviors. Future research directions are also discussed. The paper concludes with the proofs.

2. Statistical model. We consider a regression or classification problem from which we observe a sample of i.i.d. training observations $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), ..., (\mathbf{X}_n, Y_n)\}$, where each $\mathbf{X}_i = (X_i^{(1)}, ..., X_i^{(p)})^T$ denotes a set of p variables from a feature space $\mathcal{X}$. For the regression problem, Y is a real valued outcome with $E(Y^2) < \infty$; and for the classification problem, Y is binary outcome that takes values of 0 or 1. We also assume that the expected value $E(Y|\mathbf{X})$ is completely determined by a set of $p_1 < p$ variables. We refer to these p_1 variable as "strong variables", and refer to the remaining $p_2 = p - p_1$ variables as "noise variables". Without loss of generality, we assume that the strong variables are the first p_1 variables, which means $E(Y|\mathbf{X}) = E(Y|X^{(1)}, X^{(2)}, ..., X^{(p_1)})$. The goal is to consistently estimate the function $f(x) = E(Y|\mathbf{X} = x)$ and derive asymptotic properties for the estimator. To facilitate later arguments, we use $\mathcal{P}$ to denote the set $\{1, 2, ..., p\}$.

3. Reinforcement learning trees. In short, the proposed reinforcement learning trees are traditional random forests with a special type of splitting variable selection and noise variable muting at each internal node. These features are made available by implementing a reinforcement learning mechanism. Let us first consider an example which demonstrates the impact of reinforcement learning: Assume that $E(Y|\mathbf{X}) = I(X^{(1)} > 0.5)I(X^{(2)} > 0.5)$, so that $p_1 = 2$ and $p_2 = p-2$. The difficulty in estimating this structure with conventional random forests is that neither of the two strong variables show marginal effects. The immediate reward, i.e. reduction in prediction errors, from splitting on these two variables is identical to the reward obtained by splitting on one of the noise variables. Hence, it unlikely that, when p is relatively large, either $X^{(1)}$ or $X^{(2)}$ would be chosen as the splitting variable. However, if we know in advance that splitting on either $X^{(1)}$ or $X^{(2)}$ would

yield significant rewards down the road for later splits, we could confidently force a split on either variable regardless of the immediate rewards.

How we identify the most important variable at any internal node is to first fit at that node an embedded random forest and acquire the associated variable importance measures for all the covariates. Then we proceed to split the node using the most important variable(s). When doing this recursively for each daughter node, we can focus the splits on the variables which will be very likely to lead to a tree yielding the smallest prediction error in the long run.

Unfortunately, since the sample size shrinks as we move towards a terminal node, it becomes increasingly difficult to identify the important variables regardless of what embedded model we are using. On the other hand, since we have variable importance information in all the splits from the root node down to a terminal node, we should have a good idea about which variables are strong and which are not. Therefore, we will utilize this information to progressively mute noise variables during the tree construction process and to gradually restrict the search for splitting variables within a subspace of the entire feature space as the internal node sample sizes get smaller.

The remainder of this section is structured as follows: We first give a higher level algorithm outlining the main features of the RLT method in Section 3.1 without specifying the definitions of the subcomponents: embedded model, variable importance, variable deletion, and high-dimensional split. Detailed definitions of these components are given in subsequent subsections. In Sections 3.2 and 3.3 we give details of how to fit the embedded model and calculate variable importance at each internal node. In Section 3.4, we introduce a variable screening method that progressively mutes noise variables at each internal node. In Section 3.5, we extend one dimensional splits to high-dimensional splits by utilizing the available variable importance information at each internal node.

3.1. *Reinforcement learning trees.* RLT construction still follows the general pattern for an ensemble of binary trees: we first draw bootstrap samples to fit trees and then average. To construct a binary tree, a splitting variable and a splitting value is determined at each internal node, starting from the root node. This internal node is then split into two daughter nodes by grouping the observations using the selected variable and splitting value. The algorithm stops when the node sample size is sufficiently small. The key ingredient of RLT is the selection of splitting variable and also the method of constructing daughter nodes. These special features are carried out using the embedded model and variable importance measures. Table 1 summarizes

the RLT algorithm.

TABLE 1
Algorithm for reinforcement learning trees

1.	Draw M bootstrap samples from D.
2.	For the m-th bootstrap sample, where $m \in \{1, ..., M\}$, fit one RLT model $\widehat{f}_m$, using the following rules:
a)	At an internal node A, fit an embedded model $\widehat{f}_A^*$ to the data in A, restricted to the set of variables $\{1, 2, ..., p\} \backslash \mathcal{P}_A^d$, i.e. $\mathcal{P} \backslash \mathcal{P}_A^d$, where where $\mathcal{P}_A^d$ is the set of muted variables at the current node A. Details are given in Section 3.2.
b)	Using $\widehat{f}_A^*$, calculate the variable importance measure $\widehat{VI}_A(j)$ for each variable $X^{(j)}$, where $j \in \mathcal{P}$. Details are given in Section 3.3.
c)	Split node A into two daughter nodes using either i) or ii).
i)	For a one-dimensional split, use the variable with the largest variable importance measure, namely $\arg\max_j \widehat{VI}_A(j)$, as the splitting variable. The cut point c is chosen randomly and uniformly. We call this method RLT1.
ii)	For a high-dimensional split, a linear combination of variables is used. Details are given in Section 3.5. We call this method RLTk, where k is the number of variables used in the linear combination.
d)	Update the set of muted variable set $\mathcal{P}^d$ for the two daughter nodes by adding the variables with the lowest variable importance measures at the current node. Details are given in Section 3.4.
e)	Apply a)–d) on each daughter nodes until node sample size is smaller than a pre-specified value n_{min}.
3.	Average M trees to get a final model $\widehat{f} = M^{-1} \sum_{m=1}^{M} \widehat{f}_m$. For classification, $\widehat{f} = I\left(0.5 < M^{-1} \sum_{m=1}^{M} \widehat{f}_m\right)$.

3.2. *Embedded model.* To assess the variable importance $\widehat{VI}_A(j)$ for each variable j at any internal node A, we must first fit an embedded model to the internal node data. Note that at the root node, where the set of muted variables $\mathcal{P}^m = \emptyset$, all variables in the set $\mathcal{P} = \{1, 2, ..., p\}$ are considered in the embedded model and their variable importance measures will be assessed. However, as we move further down the tree, some variables will be muted and $\mathcal{P}^m \neq \emptyset$, then the embedded model will be fit using only the non-muted variable set $\mathcal{P} \backslash \mathcal{P}_A^d$. For the choice of the embedded model, we use random forests (Breiman, 2001). It is not necessary that random forests be used here. Alternatively, any learning method which is verified to be consistent with a certain convergence rate, for example, purely random forests, can be used to estimate the embedded model.

Suppose we are at an internal node A in the tree building process. To be specific, when a one-dimensional split is used, any internal node A can be

expressed as a hypercube in the feature space, i.e. $A = \{(X^{(1)}, ..., X^{(p)}) : X^{(j)} \in (a_j, b_j] \subseteq [0,1], \text{for } j \in \mathcal{P}\}$. Denote the samples at this internal node as $D_A = \{(\mathbf{X}_i, Y_i) : \mathbf{X}_i \in A\}$. We fit a random forests model, denoted by $\widehat{f}_A^*$, to the internal node data D_A with only variables that are in the set $\mathcal{P} \setminus \mathcal{P}_A^d$. For convenience, we use all the default settings in Breiman (2001) for the embedded random forests. To facilitate our later arguments, we denote the number of trees in the embedded model as M^* and denote each tree as $\widehat{f}_{A,m}^*$, for $m \in (1, 2, ..., M^*)$.

3.3. *Variable importance.* Since the purpose of fitting the embedded random forests is to determine the most important variable, we need to properly define a variable importance measure $VI_A(j)$ for each variable $j \in \mathcal{P}$ at an internal node A and use the embedded model to calculate the estimate $\widehat{VI}_A(j)$. The variable importance calculation in Breiman (2001) seems to be a natural choice here since we use random forests as the embedded model. We give the formal definition of the variable importance measure in the following. In Section 4 and Appendix section, we will carefully investigate the properties of VI_A and the asymptotic properties of its estimate $\widehat{VI}_A$.

DEFINITION 3.1. *At any internal node A, denote $\tilde{X}^{(j)}$ as an independent copy generated from the marginal distribution of $X^{(j)}$ within A, the variable importance of the j-th variable within A, namely $VI_A(j)$, is defined by:*

$$\frac{E\big[\big(f(X^{(1)}, ..., \tilde{X}^{(j)}, ..., X^{(p)}) - f(X^{(1)}, ..., X^{(j)}, ..., X^{(p)})\big)^2 | A\big]}{E\big[\big(Y - f(X^{(1)}, ..., X^{(j)}, ..., X^{(p)})\big)^2 | A\big]},$$

where the $E[\cdot|A]$ is a conditional expectation defined by $E[g(Y, \mathbf{X})|A] = E[g(Y, \mathbf{X})|I(\mathbf{X} \in A)]$, for any function g.

In practice, following Breiman (2001)'s procedure, to calculate $\widehat{VI}_A(j)$ for each fitted embedded tree, we randomly permute the values of variable j in the out-of-bag (OOB) data (to mimic the independent and identical copy $\tilde{X}^{(j)}$), drop these permuted observations down the fitted tree and then calculate the resulting mean squared error (MSE) increase. Intuitively, when j is a strong variable, randomly permuting the values of $X^{(j)}$ will result in a large $\widehat{VI}_A(j)$, while randomly permuting the values of a noise variable should result in little or no increase in MSE, so $\widehat{VI}_A(j)$ should be small. Hence $\widehat{VI}_A(j)$ calculated from the embedded model can identify the variable with greatest need-to-be-split in the sense that it explains the most variation in the outcome variable Y in the current node (see Section 4). Another important property that we observe is that for all the variables in

the muted set $\mathcal{P}_A^d$, since they are not involved in the embedded model $\widehat{f}_A^*$, randomly permuting their values will not increase MSE. Hence, for $j \in \mathcal{P}_A^d$, we must have $\widehat{VI}_A(j) = 0$. Table 2 gives details on how to assess the variable importance measure based on the embedded random forest estimator $\widehat{f}_A^*$.

TABLE 2
Variable Importance

1. For the m-th tree $\widehat{f}_{A,m}^*$, $m \in (1, 2, ..., M^*)$, in the embedded model, do steps a)–c).

a. Select the corresponding m-th OOB (out-of-bag) data which consists of the data not selected in the m-th bootstrap sample.

b. Drop OOB data down the fitted tree $\widehat{f}_{A,m}^*$ and calculate mean squared error, $MSE_{A,m}$.

c. For each variable $j \in \mathcal{P} \setminus \mathcal{P}_A^d$, do the following:

i) Randomly permute the values of the j^th variable $X^{(j)}$ in the OOB data.

ii) Drop permuted OOB data down the fitted tree $\widehat{f}_{A,m}^*$, and calculate the permuted mean squared error, $PMSE_{A,m}^j$.

2. Average over M^* measurements to get the variable importance measure for variable j:

$$VI_A(j) = \frac{\sum_{m=1}^{M^*} PMSE_{A,m}^j}{\sum_{m=1}^{M^*} MSE_{A,m}} - 1$$

.

3.4. *Variable muting.* As we discussed previously, with sample size reducing rapidly towards a terminal node during the tree construction, searching for a strong variable becomes increasingly difficult. The lack of signal from strong variables can eventually cause the splitting variable selection to behave completely randomly, and then the constructed model is similar to purely random forests. Hence, the muting procedure we introduce here is to prevent some noise variables from being considered as the splitting variable. We call this set of variables the muted set. At each internal node, we force p_d variables into the muted set, and we remove them from consideration as splitting variable at any branch of this internal node. On the other hand, to prevent strong variables from being removed from the model, we set a minimal number of p_0 variables that we always keep. This set of variables are called the protected set. We give the details of their definitions in the following. Note that both the muted set and protected set will be updated for each daughter nodes after a split is done. We first take a loot at the root node, then generalize the procedure to any internal node.

At the root node: At the root node we have $A = [0,1]^p$. After selecting the splitting variable, assume that the two resulted daughter nodes are A_L and A_R. Then we sort the variable importance measures $\widehat{VI}_A(j)$ calculated from the embedded model $\widehat{f}_A^*$ and find the p_d-th smallest value within the variable set $\mathcal{P}$ denoted by $\widehat{VI}_A^{p_d}$ and the p_0-th largest value denoted by $\widehat{VI}_A^{p_0}$. Then we define:

- The muted set for the two daughter nodes: $\mathcal{P}_{A_L}^d = \mathcal{P}_{A_R}^d = \{j : \widehat{VI}_A(j) \leq \widehat{VI}_A^{p_d}\}$, i.e. the set of variables with the smallest p_d variable importance measures.
- The protected set $\mathcal{P}_A^0 = \mathcal{P}_{A_L}^0 = \mathcal{P}_{A_R}^0 = \{j : \widehat{VI}_A(j) \geq \widehat{VI}_A^{p_0}\}$, i.e. the set of variables with largest p_0 variable importance measures. Note that the variables in the protected set will not be muted in any of the subsequent internal nodes.

At internal nodes: After the muted set and protected set have been initialized at the root split, we update the two sets in subsequent splits. Suppose at an internal node A, the muted set is $\mathcal{P}_A^d$, the protected set is $\mathcal{P}_A^0$ and the two daughter nodes are A_L and A_R. We first update the protected set for the two daughter nodes by adding the splitting variable(s) into the set:

$$\mathcal{P}_{A_L}^0 = \mathcal{P}_{A_R}^0 = \mathcal{P}_A^0 \cup \{\text{splitting variable(s) at node}A\}.$$

Note that when a one-dimensional split is used, the splitting variable is simply $\arg\max_j \widehat{VI}_A(j)$, and when a high-dimensional split is used, multiple variables could be involved.

To update the muted set, after sorting the variable importance measures $\widehat{VI}_A(j)$, we find the p_d-th smallest value within the restricted variable set $\mathcal{P} \setminus \mathcal{P}_A^d \setminus \mathcal{P}_A^0$, which value is denoted $\widehat{VI}_A^{p_d}$. Then we define the muted set for the two daughter nodes as

$$\begin{aligned} \mathcal{P}_{A_L}^d &= \mathcal{P}_{A_R}^d \\ &= \mathcal{P}_A^d \cup \{j : \widehat{VI}_A(j) \leq \widehat{VI}_A^{p_d}\} \setminus \mathcal{P}_A^0. \end{aligned}$$

REMARK 3.2. *There are two tuning parameters in the muting procedure, the number of protected variables p_0 and the number of extra muted variables at each split p_d. Ideally, we want to choose $p_0 = p_1$, which is the number of strong variable, hence the strong variables can always be protected. p_d can be any positive value less than p_2, and the noise variables will all be muted*

after finitely many splits. In practice, since we have little information about how large p_1 is, we want to set p_0 to be a reasonable large number, say $\sqrt{p}$ for a high-dimensional situation. Our updating procedure will add a strong variable into the protected set when it is used as a splitting variable. p_d dose not need to be a fixed number. It can vary depending on $|\mathcal{P} \setminus \mathcal{P}_A^d|$, which is the number of nonmuted variables at each internal node. In Section 5 we will evaluate different choices for p_d such as 0 (no muting), $20\% \cdot |\mathcal{P} \setminus \mathcal{P}_A^d|$ (moderate muting, which is suitable for most situations), and $50\% \cdot |\mathcal{P} \setminus \mathcal{P}_A^d|$ (very aggressive muting).

3.5. *High-dimensional cuts.* Using a linear combination of several variables to construct a splitting rule was considered in Breiman (2001). However, the idea never achieved much popularity. The major difficulty is computational intensity. Exhaustively searching for a linear combination of $k < p$ variables means computing and comparing approximately n^k different splitting possibilities (any k dimensional cut can be defined by k points in the feature space, and there can be as many as $n(n-1)\cdots(n-k)$ possible ways to select these k points: when n is large, this is approximately n^k). By further considering the possibility of drawing k from p total variables, it seems that the computational burden overshadows the benefit.

However, the proposed reinforcement learning splitting variable selection approach reopens the possibility of a high-dimensional split. We develop our proposed high-dimensional cut based on the following two facts. First, the splitting rule should only involve important variables. Second, the magnitude of coefficients in the linear combination should be positively related to the variable importance measure. This means that if we view the linear combination as an axis in a high-dimensional space, the axis should lean more towards the strong variables (with large variable importance) and be almost orthogonal to the noise variables (with zero variable importance).

Before presenting the algorithm for the high-dimensional cut, we define two parameters that we use to control the complexity of a high-dimensional cut:

- k: The maximum number of variables considered in the linear combination. Note that when $k = 1$, this simplifies to the usual one dimensional cut.
- α: The minimal variable importance, taking values in $(0, 1)$, of each variable in this linear combination in terms of the percentage of maximum $\widehat{VI}$ at the current node. For example, if $\alpha = 0.5$ and $max_j(\widehat{VI}(j)) = 10$ at the current node, then any variable with $\widehat{VI}$ less than 5 will not be considered for the linear combination. The purpose of this param-

eter is to ensure that the high-dimensional cut does not involve noise variables.

The high-dimensional split focuses on creating a linear combination of the form $\mathbf{X}^T\boldsymbol{\beta}$, which can be viewed as a high-dimensional axis, where $\boldsymbol{\beta}$ is a coefficient vector with dimension $p\times 1$. We can then project each observation onto this axis to provide a scalar ranking for splitting. We first give the definition of $\widehat{\boldsymbol{\beta}}_j(A)$ for each $j \in \{1, ...p\}$ at node A:

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_j(A) &= \widehat{VI}_A(j) \cdot I[\widehat{VI}_A(j) > 0] \cdot I[\widehat{VI}_A(j) \geq \widehat{VI}_A^{(k)}] \\ &\quad \cdot I[\widehat{VI}_A(j) \geq \alpha \cdot max_j \widehat{VI}_A(j)] \cdot sign(\rho_{X^{(j)},Y}(A)),\end{aligned}$$

where $\rho_{X^{(j)},Y}(A)$ is the Pearson's correlation coefficient between $X^{(j)}$ and Y within node A.

Now we give the details of each component in $\widehat{\boldsymbol{\beta}}_j(A)$. The first component is simply the variable importance measure of $X^{(j)}$. The second to the fourth component set restrictions based on the value of $\widehat{VI}_A(j)$, so that $\widehat{\boldsymbol{\beta}}_j(A)$ is non-zero only if: $\widehat{VI}_A(j)$ is positive, larger or equal to the k-th largest $\widehat{VI}$ in the current node, and larger than $\alpha \cdot 100\%$ of the largest $\widehat{VI}$ in the current node. These restrictions will eliminate all muted variables and the variables with small $\widehat{VI}$. The last component sets the sign of $\widehat{\boldsymbol{\beta}}_j(A)$ so that variables with the same trend have the same sign.

After having each $\widehat{\boldsymbol{\beta}}_j(A)$, we can calculate $\mathbf{X}_i^T\widehat{\boldsymbol{\beta}}(A)$ for each observation $\mathbf{X}_i$ in the current node. This is precisely the scalar projection of each observation for ranking mentioned above. We then select a random uniform splitting value c for this projection to separate the current node into two daughter nodes: $\{i : \mathbf{X}_i^T\widehat{\boldsymbol{\beta}}(A) \leq c, \mathbf{X}_i \in A\}$ and $\{i : \mathbf{X}_i^T\widehat{\boldsymbol{\beta}}(A) > c, \mathbf{X}_i \in A\}$.

4. Theoretical results. In this section, we develop large sample theory for the proposed RLT method. We only focus on the proof for one-dimensional splits (RLT1) in a regression problem with fixed muting parameters p_d, and we assume that the number of protected variable p_0 is larger than p_1, the number of strong variables. The main results are Theorem 4.7 which bounds below the probability of using strong variables as the splitting rule, and Theorem 4.8 which establisheds consistency and derives an error bound for RLT1. We assume, for convenience in the proofs, that the covariates $\mathbf{X}$ are generated uniformly from the feature space $\mathcal{X} = [0, 1]^p$. First, we need several other key assumptions.

ASSUMPTION 4.1. *There exist a set of strong variables* $\mathcal{S} = (1, ..., p_1)$ *such that* $f(X) = E[Y|\mathbf{X}] = E[Y|X^{(j)}, j \in \mathcal{S}]$ *and* $P\left(\frac{\partial f}{\partial X^{(j)}} = 0\right) = 0$ *for*

$j \in \mathcal{S}$. The set of noise variables is then $\mathcal{S}^c = (p_1 + 1, ..., p)$. The true function f is Lipschitz continuous with Lipschitz constant c_f.

REMARK 4.2. *The requirements for the distribution of $\mathbf{X}$ and the feature space seem restrictive, however, for any distribution with independent marginals, we can transform the distribution into the required multivariate uniform distribution. A direct consequence of this assumption is that, due to the construction of the splitting rules, any internal node can be now viewed as a hypercube in the feature space $\mathcal{X}$, i.e. any internal node $A \subseteq [0,1]^p$ has the form*

$$\{(X^{(1)}, ..., X^{(p)}) : X^{(j)} \in (a_j, b_j] \subset [0,1], \text{ for } j \in 1, ..., p\}. \tag{4.1}$$

Through out the rest of this paper, we will use the terms "internal node" and "hypercube" interchangeably provided that the context is clear.

We need to precisely define how "strong" a strong variable is, not only globally, as we did in Definition 4.1, but also locally at any internal node A. Thus we have the following assumption for the lower bound of variable importance:

ASSUMPTION 4.3. *For any hypercube A defined in the form of Equation 4.1 with the property that, for any strong variable j, $\min_{i \in \{\mathcal{S} \setminus j\}} (b_i - a_i) \geq \delta > 0$, there exist positive valued monotone functions $\psi_1(\delta)$ and $\psi_2(b_j - a_j)$, such that the variable importance of any strong variable j is bounded below by*

$$\frac{VI_A(j)}{\psi_2(b_j - a_j)} \geq \psi_1(\delta), \tag{4.2}$$

where $VI_A(j)$ is as defined in Definition 3.1.

REMARK 4.4. *This assumption basically requires that the surface of f can not be extremely flat, however, this does not require a lower bound on $\left|\partial f / \partial X^{(j)}\right|$. It is easy to verify Assumption 4.3 for a linear model, since the variable importance of a strong variable j does not depend on the interval length of other variables. In this case, we have $\psi_1(\delta) \equiv 1$ and $\psi_2(b_j - a_j) = (b_j - a_j)^2$. If f is a polynomial function with any kind of interaction, for small values of δ and $b_j - a_j$, $\psi_1(\delta)$ and $\psi_2(b_j - a_j)$ can be approximated by polynomial functions δ^{ζ_1} and $(b_j - a_j)^{\zeta_2}$, where ζ_2 is the lowest order of $X^{(j)}$ in f, and ζ_1 is the lowest order of all other variables in the interaction.*

ASSUMPTION 4.5. *With $f(\mathbf{X})$ being the true underlying function, the observed value are $Y_i = f(\mathbf{X}_i) + \epsilon_i$, where the ϵ_is are i.i.d. with mean 0 and variance σ^2. Moreover, the following Bernstein condition on the moments of ϵ is satisfied:*

$$E(|\epsilon|^m) \leq \frac{m!}{2} K^{m-2},\ m = 2, 3, ..., \tag{4.3}$$

for some constant $1 \leq K < \infty$.

Another assumption is on the embedded model. Although we use random forests as the embedded model in practice, we do not want to rule out the possibility of using any other kinds of embedded models. Hence we make the following assumption for the embedded model, which is at least satisfied for purely random forests:

ASSUMPTION 4.6. *The embedded model $\widehat{f}^*$ fitted at any internal node A with internal sample size n_A is uniformly consistent with an error bound: there exist some fixed constant $0 < K < \infty$ so that for any $\delta > 0$, $P\left(|\widehat{f}^* - f| > \delta \middle| A\right) \leq C \cdot e^{-\delta \cdot n_A^{\eta(p)} \cdot K}$, where $0 < \eta(p) \leq 1$ is a function of the dimension p, and the conditional probability on A means that the expectation is taken within the internal node A. Note that it is reasonable to assume that $\eta(p)$ is a non-increasing function of p since larger dimensions should result in poorer fitting. Furthermore, we assume that the embedded model $\widehat{f}^*$ lies in a class of functions $\mathcal{F}$ with finite entropy integral under the $L^2(P)$ norm (van der Vaart and Wellner, 1996).*

Now we present two key results, Theorem 4.7 and Theorem 4.8. Theorem 4.7 analyzes the asymptotic behavior of the variable importance measure and establisheds the probability for selecting the true strong variables and muting the noise variables. For simplicity, we only consider the case that one RLT1 tree is fitted to the entire dataset, i.e $M = 1$ and the bootstrap ratio is 100%. For the embedded model, we fit only one tree using half of the data and calculate the variable importance using the other half. We set the minimum sample size for each terminal node in RLT1 to be n^γ where $0 < \gamma < 1$. At each internal node, the splitting point c is chosen uniformly between the q-th and $(1-q)$-th quintile of each variable, where $q \in (0, 0.5]$. The smaller q is, the more diversity it induces. When $q = 0.5$, this degenerates into a model where each internal node is always split into two equally sized daughter nodes.

THEOREM 4.7. *For any internal node $A \in \mathcal{A}_n$ with sample size n_A, where $\mathcal{A}_n$ is the set of all internal nodes in the constructed RLT, define $\widehat{j}_A$ to be the selected splitting variable at A and let p_A denote the number of non-muted variables at A. Then, under Assumptions 4.1, 4.3, 4.5, 4.6, we have,*

a. $P\left(\widehat{j}_A \in \mathcal{S}\right) \geq 1 - C_1 e^{-\psi_1(\frac{n_A}{n})\cdot\psi_2(\frac{n_A}{n})\cdot n_A^{\eta(p_A)}/K_1}$, *i.e. with probability close to 1, we always select a strong variable as the splitting variable.*
b. $P\left(VI_A(\widehat{j}_A) > 2\cdot\max_j VI_A(j)\right) \geq 1 - C_2 e^{-\psi_1(\frac{n_A}{n})\cdot\psi_2(\frac{n_A}{n})\cdot n_A^{\eta(p_A)}/K_2}$, *i.e. for any internal node in the constructed RLT model, the true variable importance measure for the selected splitting variable is at least half of the true maximum variable importance with probability close to 1.*
c. *The protected set $\mathcal{P}_A^0$ contains all strong variables, i.e. $P\left(\mathcal{S} \in \mathcal{P}_A^0\right) > 1 - C_3 e^{n^{\eta(p)}/K_3}$.*

Note that in the above three results, ψ_1, ψ_2, and the constants C_k and K_k, $k = 1, \ldots, 3$, do not depend on p_A or the particular choice of A.

As we discussed in Remark 4.4, for any polynomial function, $\psi_1(\delta)$ and $\psi_2(b_j - a_j)$ can be approximately represented by δ^{ζ_1} and $(b_j - a_j)^{\zeta_2}$. Since $n_A > n^\gamma$, we have $n_A/n > n^{\gamma-1}$. Hence, to have the probability in Theorem 4.7 converging to 1, since our model eventually only involves p_0 variables, we need to tune the terminal node size parameter γ such that $n^{(\gamma-1)\zeta_1} \cdot n^{(\gamma-1)\zeta_2} \cdot n^{\gamma\eta(p_0)} \to \infty$, which requires that $\gamma > \frac{\zeta_1+\zeta_2}{\zeta_1+\zeta_2+\eta(p_0)/2}$. For a linear model, we only need $\gamma > 2/(2+\eta(p_0))$. However, in some worst case scenarios where f is relatively flat, γ has to be close to 1. This is in fact a very intuitive result because if, for example, $f = (X^{(j)})^{100}$, then we need a much longer interval over $X^{(j)}$ to detect a positive variable importance measure.

To show consistency and an error bound for RLT, we verify that the entire RLT is constructed using only strong variables provided γ is properly chosen, and that the total variation can be bounded by the variable importance measures at each terminal node, which converges to zero eventually. The most important result at this juncture is to show that the splitting variable selection process shrinks the strong variable interval length to zero at all terminal nodes. On the other hand, the variable muting mechanism relaxes the choice of γ so that it only depends on p_0 rather than p, hence the error bound for RLT only depends on p_0. To show this property, we separate the constructed RLT into two parts: the first (upper) part of the tree consists of all internal nodes with sample size larger than n^{γ^*}, where γ^* is a value between 0 and 1 such that $\psi_1(n^{\gamma^*-1}) \cdot \psi_2(n^{\gamma^*-1}) \cdot n^{\gamma^*\eta(p)} \to \infty$. Within this part of the tree, all noise variables are gradually muted so that only p_0

protected variables, which include all strong variables, remain active in each node. Note that γ^*, unlike the terminal node size parameter γ, is not a tuning parameter, but is an endogenous value determined by the true function f, the embedded model convergence rate η, and p. By the properties of ψ_1, ψ_2 and η, γ^* must be larger than γ. The second (lower) part part of the tree consists of all subsequent nodes with sample size smaller than n^{γ^*}. Since in these nodes, the embedded model only involves the p_0 protected variables, we only need to tune γ such that $\psi_1(n^{\gamma-1})\cdot\psi_2(n^{\gamma-1})\cdot n^{\gamma\eta(p_0)} \to \infty$, implying that γ depends only on $\eta(p_0)$, and thus the convergence rate for RLT only depends on p_0 and not p.

THEOREM 4.8. *Under Assumptions 4.1, 4.3, 4.5, and 4.6,* $E\left[(\widehat{f} - f)^2\right] = O_p(n^{-C})$*, where* C *is a constant that depends only on* γ*,* q*, and* p_1*. Moreover,* C *is a strictly monotone decreasing function in* p_1*.*

5. Numerical studies.

5.1. *Competing methods and parameter settings.* We compare our method with several major competitors, including the linear model with lasso, as implemented in the R package "glmnet" (Friedman et al. 2008); random forests (Breiman 2001), as implemented in the R package "randomforest"; gradient Boosting (Friedman 2001), as implemented in the R package "gbm"; and Bayesian Additive Regression Trees (Chipman et al. 2008), as implemented in the R package "BayesTree". We also include another interesting version of random forests (RF2), which fits the model, selects a set of most important variables, and refits using only these variables. For our proposed reinforcement learning trees (RLT), we include nine different versions, consisting of combinations of different tuning parameter values. The details for all simulation settings are given in the following Table 3:

5.2. *Simulation scenarios.* We create four simulation scenarios that represent different aspects which usually arise in machine learning. Such aspects include size of dimension, correlation between variables, and non-linear structure. For each scenario, we generate 200 training samples to fit the model and 1000 test samples to calculate the prediction mean squared error (MSE). Each simulation is repeated 200 times, and the averaged MSE is presented. We now describe each of our simulation settings in the following:

Scenario 1: Classification with small p. Set $p = 10$, and draw X_i independent uniforms from $[0,1]^p$. Set $\mu_i = \Phi(10 \times (X_{i,1} - 1) + 20 \times |X_{i,2} - 0.5|)$, where Φ denotes the standard normal $c.d.f$. Draw Y_i independently from $binomial(\mu_i)$.

TABLE 3
Parameter settings

Lasso	10-fold cross-validation is used with $\alpha = 1$ for lasso and λ is set to minimize cross-validation error.
Boosting	10-fold cross-validation is used. Number of trees = 3000. Optimal number of boosting iterations is determined by cross-validation.
BART	All settings are default except, when $p \geq n$, the naive estimator $\widehat{\sigma}$ is used (as implemented in Chipman et al. 2008).
RF	All settings are default.
RF2	Select the top $\sqrt{p}$ important variables from a single random forests model and refit.
RLTk	$M = 50$ trees are fit to each RLT model. We consider $k = 1, 2, 5$, namely RLT1, RLT2 and RLT5. For each of these models, as mentioned in Remark 3.2, we consider no muting ($p_d = 0$), moderate muting ($p_d = 20\% \cdot \lvert\mathcal{P} \setminus \mathcal{P}_A^d\rvert$ at any node A), and aggressive muting ($p_d = 50\% \cdot \lvert\mathcal{P} \setminus \mathcal{P}_A^d\rvert$ at any node A). To be on par with RF2, we set the number of protected variables p_0 to be $\sqrt{p}$. We also set terminal node size $n_{min} = n^{\frac{1}{3}}$.

Scenario 2: Non-linear model with correlated covariance. Set $p = 100$. To impose correlation, draw Z_i and R_i as independent uniforms from $[0, 0.8]^p$ and $[0, 0.2]$, respectively. Set the covariate vector $X_i = (Z_{i,1} + R_i, Z_{i,2} + R_i, ..., Z_{i,p} + R_i)$ and $Y_i = 10sin(\pi X_{i,1} X_{i,2}) + 20(X_{i,3} - 0.5)^2 + \epsilon_i$, where the ϵ_i are i.i.d $N(0, 1)$.

Scenario 3: Strong correlation and no marginal effect. Set $p = 100$, and draw X_i independently from $N(\mathbf{0}_{p\times 1}, \Sigma_{p\times p})$, where $\Sigma_{i,j} = \rho^{|i-j|}$ and $\rho = 0.5$, and $Y_i = 5(X_{i,10} X_{i,30}) + \epsilon_i$, where the ϵ_i are i.i.d $N(0, 1)$.

Scenario 4: linear structure with strong correlation and large p. Set $p = 300$, and draw X_i independently from $N(\mathbf{0}_{p\times 1}, \Sigma_{p\times p})$. To increase correlation, we set $\Sigma_{i,j} = \rho^{|i-j|} + 0.2 \cdot I_{(i\neq j)}$ and $\rho = 0.5$, and $Y_i = 5(X_{i,10} + X_{i,20} + X_{i,30}) + \epsilon_i$, where the ϵ_i are i.i.d $N(0, 1)$.

The first three scenarios all contain some non-linear effects which would not be captured by the Lasso. Hence we expect the Lasso to perform worse compared to other tree-based methods. However in Scenario 4, we expect the lasso to perform best due to the underlying linear model. Also, under such a linear structure, RLT2 and RLT5 should perform better than RLT1 since the linear combination split can utilize the samples in a much more efficient way. In all scenarios, we expect RF2 to perform better than RF since the number of strong variables is always less than $\sqrt{p}$, and thus the variable selection done in RF2 should be beneficial.

5.3. *Simulation results.* Table 4 summarizes testing sample MSE for each simulation setting. In Figure 1, we choose three RLT methods, RLT1 with no muting, RLT2 with moderate muting and RLT5 with aggressive muting, to plot against competing methods. There is clear evidence that under almost all settings, the proposed splitting variable selection, high-dimensional cut, and variable muting procedures all work individually and also work in combination. In general, the results show preference towards RLTk methods in general, although the method falls behind the Lasso for the linear model, which is expected. RLTk methods show advantages over all competing methods on capturing the non-linear effects in scenarios 1, 2 and 3. Scenario 3 provides an interesting illustration of how the splitting variable selection works, as is shown by RLT1 under no muting, where the MSE is reduced by up to 60.0%. When there are no marginal effects, and when the dimension is reasonably high, none of the competing methods seem to be able to capture a clear pattern. Even by reducing the dimension from 100 to $\sqrt{100} = 10$, as is done in RF2, random forests produce large MSEs. However, a slight signal in the variable importance measure from the embedded random forests can push the splits onto strong variables and improve the performance.

The improvement obtained from high-dimensional splits is also profound. In linear models, utilizing high-dimensional splits can yield huge improvements over RLT1 especially when no muting is implemented. The MSE reduction obtained by going from RLT1 to RLT5 is 39.0% (under no muting) in scenario 4. The reason is that under such a structure, linear combination splits cut the feature space more efficiently. When there is no linear combination structure, a high-dimensional split may not always be beneficial. As can be seen in scenario 3, although RLT's are significantly better than competing methods, both RLT2 and RLT5 perform slightly worse than RLT1. However, the decrease in performance is slight because of the "α" parameter enforced in the splitting process. The resulting threshold on variable importance prevents too many noise variables from being employed in the linear combination split.

When comparing different muting procedures, we also see interesting results. In scenarios 1, 2 and 4, more aggressive muting procedures improve the performance of RLT regardless of whether high-dimensional splits are implemented. In scenario 4, the MSE is reduced by 38.9%, when going from no muting to aggressive muting for RLT1, and by 29.9%, when going from no muting to moderate muting for RLT1. An interesting case is scenarios 3, where the muting procedure harms the performance, although the performance is still better than competing methods. Note that in scenario 3, a setting with no marginal effect and only two strong variables, a very aggres-

sive muting procedure appears to mute the strong variables early on so that they are ruled out from the model. Considering that the embedded model (RF) is not especially accurate in this situation, aggressive muting may not be a good choice for scenarios 3.

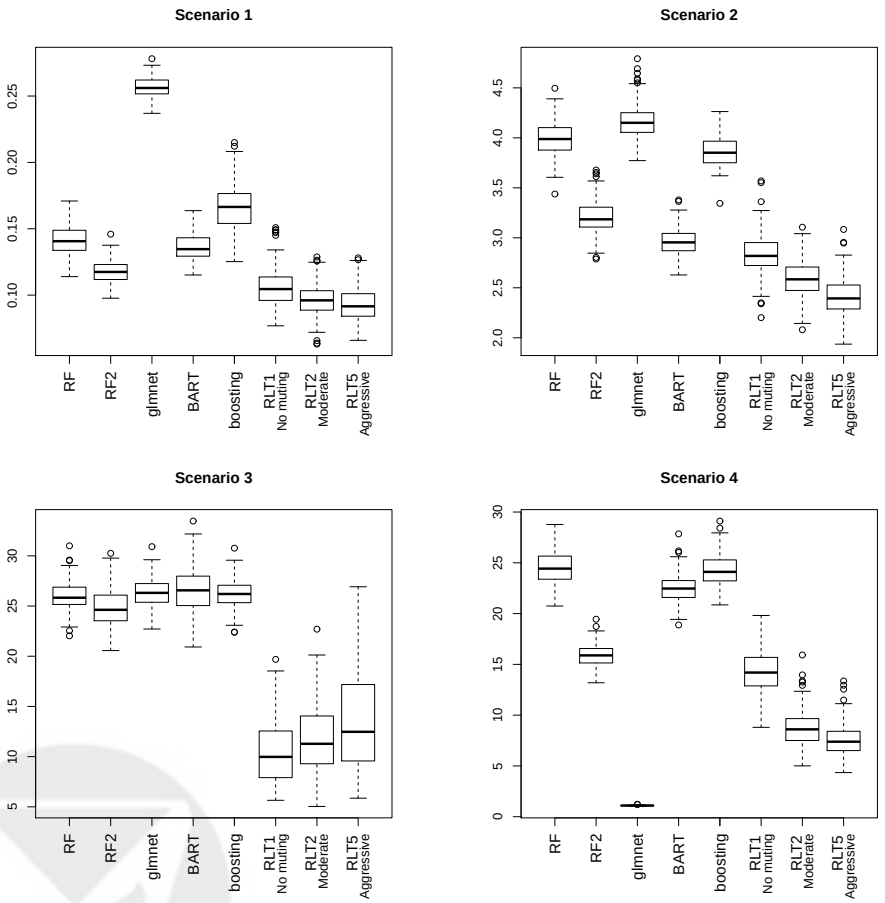
TABLE 4
Prediction Mean Squared Error

		Scenario 1	Scenario 2	Scenario 3	Scenario 4
RF		0.142	4.005	25.811	24.658
RF2		0.118	3.217	24.449	15.962
glmnet		0.257	4.191	26.100	1.099
BART		0.137	2.963	26.358	22.611
boosting		0.167	3.876	25.927	24.306
Muting	RLTk				
No	RLT1	0.106	2.831	9.774	14.271
	RLT2	0.100	2.698	10.209	9.103
	RLT5	0.101	2.706	10.421	8.709
Moderate	RLT1	0.098	2.658	11.644	10.009
	RLT2	0.096	2.593	11.938	8.682
	RLT5	0.096	2.597	11.917	8.525
Aggressive	RLT1	0.093	2.468	13.568	8.726
	RLT2	0.093	2.415	14.020	7.618
	RLT5	0.093	2.408	14.045	7.556

5.4. *Data analysis example.* The diagnostic Wisconsin breast cancer database (Mangasarian et al. 1995) has been a popular dataset for evaluating machine learning. We obtained the data from the UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/). The dataset contains diagnostic results from 569 subjects, classed as either "benign" or "malignant". A total of 30 features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The features describe characteristics of the cell nuclei present in the image, such as radius, texture, perimeter, area, etc. In our analysis of this data, we want to compare the performance of different methods and also demonstrate the impact of increased dimension on prediction error.

The original data is standardized to let each covariate have mean zero and variance one. We keep the exact same parameter settings given in section 4.1 and create an independent set of new covariates to increase the total num-

FIG 1. *Box plot of prediction Mean Squared Error*



ber of covariates p by 100, 200, 300, 400 and 500. These extra covariates are independent standard normal random deviates. We then randomly sample 300 observations without replacement from the total of 569 as the training dataset, and use the remaining observations as a testing sample to compute the misclassification rate. Due to the high dimension, this procedure is repeated 500 times and averaged to stabilize the results.

The misclassification rates are summarized in Table 5. We picked three RLT method: RLT1 with no muting (the overall worst RLT method), RLT2 with moderate muting and RLT5 with aggressive muting, and plotted them against competing methods in Figure 5.4. When only the original 30 covariates are used, glmnet performs best with a misclassification rate of 3.1%, followed by RLT5 with no muting (3.3%), all moderate muting RLT's ($3.3 \sim 3.4\%$), BART (3.8%) and RLT2 with no muting (3.8%). As the dimension reaches 530, RLT become the dominant methods with misclassification rates in the range of $3.5 \sim 3.8\%$, except RLT1 with no muting and RLT2 with no muting. glmnet (4.1%) and RF2 (4.4%) are the best two among the competing methods.

It is interesting to observe two sets of comparisons here: RLT1 with no muting vs. RF; and aggressive RLT's vs. RF2. RLT1 with no muting and RF start off with similar performance when $p = 30$. However, as the dimension increases, the reinforcement learning variable selection starts to show its benefit and eventually reduces the misclassification rate by 10.79% from RF. On the other hand, the misclassification rates for both RF2 and aggressive RLT methods decrease in this simulation. Keeping in mind that both methods will exclude a large proportion of variables, it is not surprising to see this pattern. With only 30 covariates in the initial model, RF2 will only consider the best 5 variables, and aggressive RLT's will mute, on average, 22.5 (75%) variables in the first two splits and only 5 variables are protected against muting. This causes both of them to very likely miss some true strong variables. As p increases, the methods will eventually be able to fit the model with the most strong variables included. However, aggressive RLT's are uniformly better in this comparison regardless of the implementation of high dimensional splits.

The plot also shows an important advantage of RLT: it performs consistently across changing dimension, which means that it has good immunity to dimension. While being the second best method at $p = 30$, RLT5 with moderate muting has its misclassification rate increase by only 10.35% when p is increased to 530. This is quite impressive compared to glmnet's increase of 33.64%, RF's of 24.33% and BART's of 50.58%.

FIG 2. *Misclassification rate by increasing dimension*

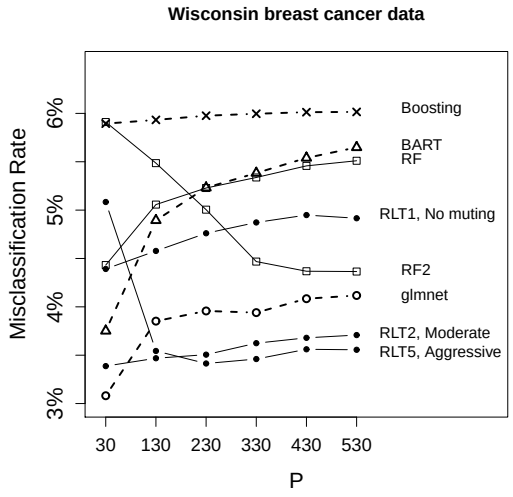


TABLE 5
Diagnostic Wisconsin Breast Cancer Dataset misclassification rate

		p=30	p=130	p=230	p=330	p=430	p=530
RF		0.044	0.051	0.052	0.053	0.055	0.055
RF2		0.059	0.055	0.050	0.045	0.044	0.044
glmnet		0.031	0.039	0.040	0.039	0.041	0.041
BART		0.038	0.049	0.052	0.054	0.055	0.056
Boosting		0.059	0.059	0.060	0.060	0.060	0.060
Muting	RLTk						
No	RLT1	0.044	0.046	0.048	0.049	0.049	0.049
	RLT2	0.038	0.040	0.041	0.042	0.043	0.043
	RLT5	0.033	0.036	0.037	0.038	0.039	0.038
Moderate	RLT1	0.034	0.035	0.036	0.037	0.038	0.038
	RLT2	0.034	0.035	0.035	0.036	0.037	0.037
	RLT5	0.033	0.034	0.035	0.036	0.037	0.037
Aggressive	RLT1	0.051	0.036	0.035	0.035	0.036	0.036
	RLT2	0.051	0.035	0.034	0.035	0.036	0.036
	RLT5	0.050	0.035	0.034	0.035	0.035	0.035

5.5. *Numerical study conclusion.* In this numerical study section, we compared the performance of the proposed RLT method with several popular learning tools. Under both simulated scenarios and the Wisconsin Breast Cancer Dataset, the results favor RLT methods. There is a significant improvement over competing methods in most situations, however, the results vary some depending on the choice of tuning parameters. RLT methods with moderate muting generally perform the best and most stably across different settings, and incorporating high dimensional splits seems almost always beneficial. On the other hand, when the dimension is relatively low, aggressive muting can sometimes cause strong variables to be muted and harm the performance; when the dimension is high, aggressive muting starts to show a noticeable benefit. The behavior of different muting procedures needs further analysis, and we do not suggest using aggressive muting, unless the dimension is very high, due to its apparent instability in low-dimensional situation.

6. Discussion. We proposed reinforcement learning trees in this paper. By fitting an embedded random forest model at each internal node, and calculating the variable importance measures, we can increase the chance of selecting the most important variables to cut and thus utilize the available training samples in an efficient way. The proposed high-dimensional splitting strategy extends the use of variable importance measures and creates splitting rules based on a linear combination of variables. The variable muting procedures further concentrates the splits on the strong variables at deep nodes in the tree where the node sample size is small. All of these procedures take advantage of Reinforcement Learning and yield significant improvement over existing methods especially when the dimensional is high and the true model structure is sparse. There are several remaining issues we want to discuss in this section including the choice of tuning parameters, computational issue, and future research directions.

6.1. *Choosing the tuning parameters.* The number of trees M in RLT does not need to be very large to achieve good performance. In all simulations, we used $M = 50$. The use of high-dimensional splits (RLT2 and RLT5) seems beneficial in most situations, and the drawbacks are negligible even when there is no linear effect. Hence we recommend choosing $k = 2$ to 5 and using $\alpha = 0.5$. In all simulations, we use terminal node size equal to $n^{1/3}$ which seems to perform reasonably well. However, the optimal choice of γ needs further theoretical analysis. The choice for muting parameters seems tricky. Ideally, the choice of p_0 and p_d should depend on sample size n, dimension p, and even the performance of the embedded model, which

can be hard to evaluate. In general, we recommend using a moderate muting procedure, i.e., $p_d = 20\% \cdot |\mathcal{P} \setminus \mathcal{P}_A^d|$ at each internal node, and using $p_0 = \sqrt{p}$. However, the choice of these parameters is flexible and should depend on the setting. For example, when p is extremely large, a more aggressive muting procedure should probably be used to force a sparse structure. These adjustments require testing on a massive number of datasets and will be a focus area for our future research.

6.2. *Computational intensity.* The computational cost of RLT is higher than the original random forests, which is expected since more computations need to be done at each internal node to search for the optimal splitting variable. In a worst case scenario, RLT will fit as many as $n^{1-\gamma}$, $0 < \gamma < 1$ random forests if we require the terminal sample size to be at least n^γ. However, this is not entirely necessary because as splitting moves towards a terminal node, the sample size shrinks rapidly and will not require as much computation as needed at root nodes. Hence, the number of trees in the embedded model can decrease as the internal node sample size decreases. Moreover, the muting procedure eliminates a large proportion of variables so that the embedded model takes less time to fit. On the other hand, RLT carries out high-dimensional splitting at little extra computational cost, which compared to exhaustive searching, is much less computationally intensive. In our simulation study, where the method is implemented in R2.14, RLT with aggressive muting usually takes 100 times the computational time of random forests. We plan to implement our method in C and incorporate parallel computing to speed up the computation. Also we plan to test different embedded models, such as Extremely Randomized Trees (Geurts et al., 2006), to reduce the computational burden.

6.3. *Future research.* Our theoretical results established the consistency of RLT1 and show that the error rate only depends on p_0. However, achieving a tight error bound for this splitting mechanism and comparing it to other types of models seems to be nearly insurmountable at the current stage. This seems to be because the optimal choice of terminal node size γ depends on the behavior of the embedded model, and even depends on the true underlying function f. However, it needs to be verified that choosing a smaller terminal node size will not negate the dependence on p_0 and make the model inconsistent. This can be seen by considering the worst case scenario where we randomly select the splitting variable after the node size n^γ is reached. Our next step for future research is to extend the existing structure to a simpler embedded model and relatively restricted underlying function in the hope of explicitly deriving a tight error bound. It is known

(Biau et al. (2008)) that forest averaging can potentially improve the performance of tree-based model, so it is also of interest to study the impact of forest averaging for RLT. Another direction we are currently working on is to find other techniques to improve the error rate so that it only depends on p_1 rather than on a pre-selected p_0. Again, this may involve a relatively restricted underlying function f.

Implementing the Reinforcement Learning mechanism into other types of tree-based methods can also be an interesting research direction. For example, in a recent paper Zhu and Kosorok (2012) developed a censoring imputation technique (RIST) to improve random forests in censored data settings. If Reinforcement Learning is transplanted to the RIST model, the accuracy of the censoring imputation step should be notably increased, so that it could further push the refitted model to concentrate only on strong variables. The Reinforcement Learning mechanism introduces a significant benefit which could potentially be enjoyed by not only tree-based methods, but also by other non-parametric modeling approaches. We plan on pursuing research in these directions.

7. Appendix.

Proof of Theorem 4.7.

Step 1: We first establish the asymptotic results for the variable importance measure. Without further specification, the proof of Step 1 is conditional on an internal node A with sample size n_A and number of non-muted variables equal to p_A. We denote the internal node dataset by $\mathcal{D}_A = \{(X_i, Y_i), i \in A\}$. Let $\mathbb{P}$ be the probability measure of $((X), Y)$ and let $\mathbb{P}$ be the corresponding empirical measure.

First, we observe that, $VI_A(j)$ is bounded. By Assumption 4.1, f is Lipschitz continuous with Lipschitz constant c_f,

$$\begin{aligned} & VI_A(j) \\ &= \frac{E[E[(f(X_i^{(1)}, ..., \tilde{X}_i^{(j)}, ..., X_i^{(p)}) - f(X_i^{(1)}, ..., X_i^{(j)}, ..., X_i^{(p)}))^2 | X_i^{(j)}] | A]}{\sigma^2} \\ &\leq \frac{E[E[(c_f \cdot (b_j - a_j))^2 | X_i^{(j)}] | A]}{\sigma^2} = \frac{c_f^2 \cdot (b_j - a_j)^2}{\sigma^2}. \end{aligned} \tag{7.1}$$

Hence $VI_A(j)$ is also bounded above by the interval length of $X^{(j)}$, i.e. $(b_j - a_j)$, in A. It can be further bounded above by $\frac{c_f^2}{\sigma^2}$ since $(b_j - a_j) < 1$ for any internal node A.

Now we show that $\widehat{VI}_A(j)$ converges to $VI_A(j)$ at an exponential rate. For simplicity, assume that the embedded model $\widehat{f}_A^*$ randomly selects half of $\mathcal{D}_A$ to fit the model, denoted by $\mathcal{D}_{A_1}$, and the variable importance is calculated using the other half of the data, denoted by $\mathcal{D}_{A_2}$. Noticing that this is exactly (except for the proportion of each subset) what we do for each tree in a standard random forests model. However, with the potential use of other models, this simplifies the formulation. Further, since the j-th variable importance measure is calculated by randomly permuting the values of $X_i^{(j)}$ in $\mathcal{D}_{A_2}$, which we denote by $\tilde{X}_i^{(j)}$, we assume that this permutation is done infinitely many times. Then, for the i-th observation in $\mathcal{D}_{A_2}$, the squared error after permutation is $E_{\tilde{X}_i^{(j)}}\big(\widehat{f}_A^*(X_i^{(1)},...,\tilde{X}_i^{(j)},...,X_i^{(p)})-Y_i\big)^2$. Hence the j-th variable importance can be formulated as:

$$\begin{aligned}
&\widehat{VI}_A(j)\\
&= \frac{\frac{1}{n_A/2}\sum_{\mathbf{X}_i\in\mathcal{D}_{A_2}} E_{\tilde{X}^{(j)}}\Big(\widehat{f}_A^*(X_i^{(1)},...,\tilde{X}^{(j)},...,X_i^{(p)})-Y_i\Big)^2}{\frac{1}{n_A/2}\sum_{\mathbf{X}_i\in\mathcal{D}_{A_2}} E_{\tilde{X}^{(j)}}\Big(\widehat{f}_A^*(X_i^{(1)},...,X_i^{(j)},...,X_i^{(p)})-Y_i\Big)^2}-1\\
&= \frac{\frac{1}{n}\sum_{\mathbf{X}_i\in\mathcal{D}} E_{\tilde{X}^{(j)}}\Big(\widehat{f}_A^*(X_i^{(1)},...,\tilde{X}^{(j)},...,X_i^{(p)})-Y_i\Big)^2 I_{[\mathbf{X}_i\in A_2]}}{\frac{1}{n}\sum_{\mathbf{X}_i\in\mathcal{D}} E_{\tilde{X}^{(j)}}\Big(\widehat{f}_A^*(X_i^{(1)},...,X_i^{(j)},...,X_i^{(p)})-Y_i\Big)^2 I_{[\mathbf{X}_i\in A_2]}}-1,
\end{aligned} \tag{7.2}$$

where $I[\mathbf{X}_i\in A_2]$ is the indicator function denoting that $\mathbf{X}_i$ falls into the internal node A, and is randomized with probability $\frac{1}{2}$ to $\mathcal{D}_{A_2}$ for calculating variable importance. Let the set $(X_i^{(1)},...,X_i^{(j-1)},X_i^{(j+1)}...,X_i^{(p)})$ be $X_i^{(-j)}$. Then the numerator of the first term of (7.2) can be broken down into:

$$\begin{aligned}
&\frac{1}{n}\sum_{\mathbf{X}_i\in\mathcal{D}} E_{\tilde{X}^{(j)}}\Big(\widehat{f}_A^*(X_i^{(1)},...,\tilde{X}^{(j)},...,X_i^{(p)})-Y_i\Big)^2 I_{[\mathbf{X}_i\in A_2]}\\
&\quad= \mathbb{P}_n\Big(E_{\tilde{X}^{(j)}}\big(\widehat{f}_A^*(X^{(-j)},\tilde{X}^{(j)})-Y\big)^2 I_{[\mathbf{X}\in A_2]}\Big)\\
&\quad= (\mathbb{P}_n-\mathbb{P})\Big(E_{\tilde{X}^{(j)}}(\widehat{f}_A^*(X^{(-j)},\tilde{X}^{(j)})-Y)^2 I_{[\mathbf{X}\in A_2]}\Big)\\
&\qquad+\mathbb{P}\Big(E_{\tilde{X}^{(j)}}\big(\widehat{f}_A^*(X^{(-j)},\tilde{X}^{(j)})-f_A(X^{(-j)},\tilde{X}^{(j)})\big)^2 I_{[\mathbf{X}\in A_2]}\Big)\\
&\qquad+\mathbb{P}\Big(E_{\tilde{X}^{(j)}}\big(f_A(X^{(-j)},\tilde{X}^{(j)})-f_A(X^{(-j)},X^{(j)})\big)^2 I_{[\mathbf{X}\in A_2]}\Big)\\
&\qquad+\mathbb{P}\Big(E_{\tilde{X}^{(j)}}\big(f_A(X^{(-j)},X^{(j)})-Y\big)^2 I_{[\mathbf{X}\in A_2]}\Big)\\
&\quad=: \tilde{T}_1+\tilde{T}_2+\tilde{T}_3+\tilde{T}_4.
\end{aligned} \tag{7.3}$$

Now we bound each of the four terms in Equation 7.3. We will first show the bound for $\tilde{T}_1$ and then for $\tilde{T}_2^*$, following the same idea. We use Theorem 8 in van de Geer and Lederer (2011) to establish the bound for $\tilde{T}_1$. The Theorem states that for any function $g(\mathbf{X})$ that lives in a collection of functions $\mathcal{G}$, if the Bernstein condition

$$\sup_{g\in\mathcal{G}} E|g|^m \leq \frac{m!}{2}K^{m-2},\ m=2,3,... \tag{7.4}$$

is satisfied for some constant $K \geq 1$, then $\sqrt{n}(\mathbb{P}_n - \mathbb{P})g$ has exponential tail.

By Assumption 4.6, $\widehat{f}^*$ has exponential tail. On the other hand, $Y = f(\mathbf{X}) + \epsilon$, and $f(\mathbf{X})$ are bounded, and hence Y also satisfies the moment condition by Assumption 4.5. Hence, we can fine some constant K such that the following Bernstein condition is satisfied:

$$\sup_{\widehat{f}^*} E\left|f_A^*(X^{(-j)}, \tilde{X}^{(j)}) - Y\right|^m \leq \frac{m!}{2}K^{m-2},\ m=2,3,.... \tag{7.5}$$

Furthermore, since $\widehat{f}^*$ has finite entropy integral by Assumption 4.6, we can use Theorem 8 in van de Geer and Lederer (2011) and reorganize the terms to can find a constant $K_1^* > 0$ such that:

$$P\left(\sup\left|\sqrt{n}\tilde{T}_1\right| \geq t\right) \leq e^{-t/K_1^*}. \tag{7.6}$$

For $\tilde{T}_2$, we first write it into a conditional probability $\mathbb{P}_{A_2}$ such that

$$\begin{aligned}\tilde{T}_2 &= \mathbb{P}_{A_2}\left(E_{\tilde{X}^{(j)}}\left(\widehat{f}_A^*(X^{(-j)}, \tilde{X}^{(j)}) - f_A(X^{(-j)}, \tilde{X}^{(j)})\right)^2\right)P(A_2) \\ &= \tilde{T}_2^* P(A_2). \end{aligned} \tag{7.7}$$

For $\tilde{T}_2^*$, noting Assumption 4.6 for the error bound of f_A^*, and following similar arguments as applied to $\tilde{T}_1$, we have for some constant $K_2^* > 0$:

$$P\left(\sup\left|\sqrt{n_A^{\eta(p_A)}}\tilde{T}_2^*\right| \geq t\right) \leq e^{-t/K_2^*}. \tag{7.8}$$

For the other two terms, it is easy to see by Definition 3.1 that $\tilde{T}_3 = VI_A(j)\sigma^2 P(A_2)$, and $\tilde{T}_4 = \sigma^2 P(A_2)$ by Assumption 4.5.

Note that the denominator of the first term in (7.2) can be decomposed into four terms: T_1, T_2, T_3^* and T_4, similar to (7.3) but with $X_i^{(j)}$ in the lieu

of $\tilde{X}_i^{(j)}$. The first two terms can be bounded in the same way as the above. The third term equals 0 since $\tilde{X}_i^{(j)}$ is replaced by $X_i^{(j)}$. And the fourth term $T_4 = \sigma^2 P(A_2)$.

Hence, together with (7.6), (7.8) for the numerator, and the above arguments for the denominator, we can derive that

$$
\begin{aligned}
&P\Big(\Big|\widehat{VI}_A(j) - VI_A(j)\Big| > C\Big) \\
&= P\Big(\Big|\frac{\tilde{T}_1 + \tilde{T}_2^* P(A_2) + \sigma^2 P(A_2) VI_A(j) + \sigma^2 P(A_2)}{T_1 + T_2^* P(A_2) + 0 + \sigma^2 P(A_2)} - 1 - VI_A(j)\Big| > C\Big) \\
&\leq P\Big(\Big|\frac{\tilde{T}_1}{T_1 + T_2^* P(A_2) + \sigma^2 P(A_2)}\Big| > C/3\Big) \\
&\quad + P\Big(\Big|\frac{\tilde{T}_2^* P(A_2)}{T_1 + T_2^* P(A_2) + \sigma^2 P(A_2)}\Big| > C/3\Big) \\
&\quad + P\Big(\Big|\frac{\sigma^2 P(A_2)(VI_A(j)+1)}{T_1 + T_2^* P(A_2) + \sigma^2 P(A_2)} - 1 - VI_A(j)\Big| > C/3\Big) \\
&= P\Big(\Big|\frac{\tilde{T}_1}{T_1 + T_2^* P(A_2) + \sigma^2 P(A_2)}\Big| > C/3\Big) \\
&\quad + P\Big(\Big|\frac{\tilde{T}_2^* P(A_2)}{T_1 + T_2^* P(A_2) + \sigma^2 P(A_2)}\Big| > C/3\Big) \\
&\quad + P\Big(\Big|\frac{(T_1 + T_2^* P(A_2))(1 + VI_A(j))}{T_1 + T_2^* P(A_2) + \sigma^2 P(A_2)}\Big| > C/3\Big). \qquad (7.9)
\end{aligned}
$$

Noticing that all the T terms are positive, and $VI_A(j)$ is also positive and bounded above, we have:

$$
\begin{aligned}
&P\Big(\Big|\widehat{VI}_A(j) - VI_A(j)\Big| > C\Big) \\
&\leq P\Big(\Big|\frac{\tilde{T}_1}{\sigma^2 P(A_2)}\Big| > C/3\Big) + P\Big(\Big|\frac{\tilde{T}_2^* P(A_2)}{\sigma^2 P(A_2)}\Big| > C/3\Big) + \\
&\quad P\Big(\Big|\frac{T_1(1 + VI_A(j))}{\sigma^2 P(A_2)}\Big| > C/6\Big) + P\Big(\Big|\frac{T_2^* P(A_2)(1 + VI_A(j))}{\sigma^2 P(A_2)}\Big| > C/6\Big) \\
&\leq e^{-C \cdot P(A_2) \cdot n/3K_1} + e^{-C \cdot n_A^{\eta(p_A)}/3K_2} + e^{-C \cdot P(A_2) \cdot n/3K_3} + e^{-C \cdot n_A^{\eta(p_A)}/3K_4} \\
&\leq e^{-C \cdot n_A^{\eta(p_A)}/K_5}. \qquad (7.10)
\end{aligned}
$$

Noting that this is the tail probability for $\widehat{VI}_A(j)$ when p_A variables are considered in the embedded model, we can easily generalize it to the

situation at an internal node where only p_0 variables are considered. In this case, we replace $\eta(p)$ by $\eta(p_0)$, yielding a faster convergence rate. In the derivation, the constant K_5 can possibly depend on p_A, however, since $p_A < p$, which is finite, we can always choose a larger K_5 such that the equation holds for all values of p_A. Consequently, K_5 does not depend on the choice of internal node A.

Now, two situations can arise for $VI_A(j)$:

Situation 1: $X^{(j)}$ is a noise variable. Since changing the value of $X^{(j)}$ will not change $f(X)$, $f(X^{(1)}, ..., \tilde{X}^{(j)}, ..., X^{(p)}) \equiv f(X^{(1)}, ..., X^{(j)}, ..., X^{(p)})$, and thus $VI_A(j) \equiv 0$.

Situation 2: $X^{(j)}$ is a strong variable. According to Assumption 4.3, $VI_A(j)$ is bounded below by $\psi_1(\delta) \cdot \psi_2(b_j - a_j)$, where $\delta = \min_{i \in \{\mathcal{S} \setminus j\}} (b_i - a_i)$. We further note that since the internal node size is n_A, the interval length of any variable is at least $\frac{n_A}{n}$ even if all splits are made on that variable. Hence both δ and $b_j - a_j$ are larger than $\frac{n_A}{n}$. Hence $VI_A(j) \geq \psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n})$ for any strong variable.

Hence, to sum up situations (1) and (2), we have

$$VI_A(j) \begin{cases} \geq \psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n}), & \text{if } j \in \mathcal{S}. \\ = 0, & \text{if } j \in \mathcal{S}^c. \end{cases} \tag{7.11}$$

Step 2: Now we prove a) of this Theorem. Let $\widehat{j}_A$ be the selected splitting variable at internal node A, i.e. $\widehat{j}_A = \arg\max_j VI_A(j)$. Without loss of generality, we assume that at this internal node A, the true variable importance measures are in the order $VI_A(1) \geq VI_A(2) \geq \cdots \geq VI_A(p_1) > VI_A(p_1 + 1) = \cdots = VI_A(p) = 0$. Then the probability that the selected splitting variable $\widehat{j}_A^*$ belongs to the set of strong variables satisfies the following inequality:

$$\begin{aligned} & P(\widehat{j}_A \in \mathcal{S}) \\ = \; & 1 - P(\widehat{j}_A \in \mathcal{S}^c) \\ = \; & 1 - \sum_{i \in \mathcal{S}^c} P(\widehat{j}_A = i) \\ \geq \; & 1 - \sum_{i \in \mathcal{S}^c} P\big(\widehat{VI}_A(i) > \widehat{VI}_A(j), \text{ for all } \; j \in \mathcal{S}\big) \\ \geq \; & 1 - p_1 \sum_{i \in \mathcal{S}^c} P(\widehat{VI}_A(i) > \widehat{VI}_A(p_1)). \end{aligned} \tag{7.12}$$

Let $\widehat{\Delta}_j = \widehat{VI}_A(j) - VI_A(j)$. Using equation (7.10) and noting that $VI_A(i) =$

0 for all $i \in \mathcal{S}^c$, the above probability can be bounded below by

$$\begin{aligned} & P(\widehat{j}_A \in \mathcal{S}) \\ &\geq 1 - p_1 \sum_{i \in \mathcal{S}^c} P\big(\widehat{\Delta}_j + 0 > \widehat{\Delta}_{p_1} + VI_A(p_1)\big) \\ &\geq 1 - p_1 \sum_{i \in \mathcal{S}^c} \Big[P\big(|\widehat{\Delta}_{p_1}| > \frac{VI_A(p_1)}{2}\big) + P(\widehat{\Delta}_j > \frac{VI_A(p_1)}{2}) \Big] \\ &= 1 - p_1 \sum_{i \in \mathcal{S}^c} 4 \cdot e^{-\frac{VI_A(p_1)}{2} \cdot n_A^{\eta}/K_5} \\ &= 1 - 4p_1p_2 \cdot e^{-\frac{VI_A(p_1)}{2} \cdot n_A^{\eta}/K_5}. \end{aligned} \tag{7.13}$$

Using Equation 7.11, we have, for any internal node A with sample size n_A, and with p_A nonmuted variables,

$$P(\widehat{j}_A \in \mathcal{S}) \geq 1 - 4p_1p_2 \cdot e^{-\psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n}) \cdot n_A^{\eta(p_A)}/(K_5 \cdot 2)}. \tag{7.14}$$

Since p_1, p_2 and K_5 are all constant, the proof for a) is concluded.

Step 3: We show b) using a similar structure as the proof of a). Note that at any internal node A, the probability that the maximum true variable importance is larger than twice that of the selected splitting variable is

$$P\Big(\max_j VI_A(j) > 2VI_A\big(\widehat{j}_A\big) \Big).$$

By defining the variable with the true maximum variable importance at node A as $j_A^m = \arg\max_j VI_A(j)$, the above equation can be bounded by

$$\begin{aligned} & P\big(VI_A(j_A^m) > 2VI_A(\widehat{j}_A)\big) \\ &\leq P\Big(VI_A(j_A^m) > VI_A(\widehat{j}_A) + \psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n})\Big) \\ &= P\Big(VI_A(j_A^m) - \widehat{VI}_A(j_A^m) > VI_A(\widehat{j}_A) - \widehat{VI}_A(j_A^m) + \psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n})\Big) \\ &= P\Big(VI_A(j_A^m) - \widehat{VI}_A(j_A^m) > VI_A(\widehat{j}_A) - \widehat{VI}_A(\widehat{j}_A) \\ &\qquad + \widehat{VI}_A(\widehat{j}_A) - \widehat{VI}_A(j_A^m) + \psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n})\Big). \end{aligned}$$

Note that $\widehat{VI}_A(\widehat{j}_A) - \widehat{VI}_A(j_A^m) \geq 0$ since $\widehat{j}_A$ is the selected variable. Adapting the notation of $\widehat{\Delta}$ used in Step 2, we now have

$$P\Big(VI_A(j_A^m) > 2VI_A\big(\widehat{j}_A\big)\Big)$$

$$
\begin{aligned}
&\leq P\Big(\widehat{\Delta}_{j_A^m} > \widehat{\Delta}_{\widehat{j}_A} + 0 + \psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n})\Big) \\
&\leq P\Big(|\widehat{\Delta}_{j_A^m}| > \frac{\psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n})}{2}\Big) \\
&\quad + P\Big(|\widehat{\Delta}_{\widehat{j}_A}| > \frac{\psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n})}{2}\Big) \\
&\leq 4e^{-\psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n}) \cdot n_A^{\eta(p_A)}/(K_5 \cdot 2)}. \qquad (7.15)
\end{aligned}
$$

Thus the proof for b) is concluded.

Step 4: We now show c), that the protected set $\mathcal{P}_A^0$ for the entire tree contains all strong variables with probability close to 1, provided the number of protect variables p_0 is greater than p_1. It is sufficient to show this property at the root node, where $A = [0,1]^p$, since the protected set will only increase after a split. Note that when $p_0 > p_1$, if a strong variable is not in the protected set, there must be at least one noise variable with larger $\widehat{VI}$. Hence we have:

$$
\begin{aligned}
&P(\mathcal{S} \in \mathcal{P}_A^0) \\
&\geq 1 - P(\exists j \in \mathcal{S} \text{ and } i \in \mathcal{S}^c, s.t. \widehat{VI}_A(j) < \widehat{VI}_A(i)) \\
&\geq 1 - \sum_{j \in \mathcal{S}, i \in \mathcal{S}^c} P(\widehat{VI}_A(j) < \widehat{VI}_A(i)) \\
&\geq 1 - p_1 p_2 P(\widehat{VI}_A(p_1) < \widehat{VI}_A(p_1 + 1)).
\end{aligned}
$$

By similar arguments to those used in Steps 2), and noting that $n_A = n$ at the root node, we can bound the above probability by:

$$
\begin{aligned}
&P(\mathcal{S} \in \mathcal{P}_A^0) \\
&\geq 1 - p_1 p_2 e^{VI_A(p_1) \cdot n^{\eta(p)}/(K_5 \cdot 2)}. \qquad (7.16)
\end{aligned}
$$

Since at the root node, all the variable importance measures, including $VI_A(p_1)$, are fixed constants, The proof for c) is concluded.

□

Proof of Theorem 4.8. We prove this theorem in two steps. First, we show that for the entire constructed RLT, with exponential rate, only strong variables are used as splitting variables. Second, we derive consistency and error bounds by bounding the total variation using the terminal node size variable importance which converges to zero.

Step 1: In this step, we show that for the entire tree, only strong variables are used as the splitting variable, and furthermore, the variable importance

measure for the splitting variable is at least half of the maximum variable importance at each split. First, it is easy to verify that, both a) and b) in Theorem4.7 can be satisfied simultaneously with probability bounded below by

$$1 - C \cdot e^{-\psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n}) \cdot n_A^{\eta(p)}/K}. \tag{7.17}$$

Define $\mathcal{A}$ as the set of all internal nodes. Recall that $\psi_1(\delta)$ and $\psi_2(b_j - a_j)$ can be approximated by δ^{ζ_1} and $(b_j - a_j)^{\zeta_2}$, respectively. Thus we can always find a $\gamma^* < 1$ such that when $n_A > n^{\gamma^*}$, $\psi_1(\frac{n_A}{n}) \cdot \psi_2(\frac{n_A}{n}) \cdot n_A^{\eta(p)} \to \infty$. We define two groups of internal nodes $\mathcal{A}_1 = \{A_i,\, s.t.\, A_i \in \mathcal{A}, n_{A_i} \geq n^{\gamma^*}\}$ and $\mathcal{A}_2 = \{A_i,\, s.t.\, A_i \in \mathcal{A}, n_{A_i} < n^{\gamma^*}\}$, where n_{A_i} is the sample size at node A_i. Then we bound the probability:

$$\begin{aligned} & P\left(\left\{\widehat{j}_A \in \mathcal{S} \text{ and } \max_j VI_A(j) > 2VI_A\big(\widehat{j}_A\big), \text{ for all } A_i \in \mathcal{A}\right\}^c\right) \\ & \leq \sum_{A_i \in \mathcal{A}_1} P\left(\left\{\widehat{j}_{A_i} \in \mathcal{S} \text{ and } \max_j VI_{A_i}(j) > 2VI_{A_i}\big(\widehat{j}_{A_i}\big)\right\}^c\right) \\ & + \sum_{A_i \in \mathcal{A}_2} P\left(\left\{\widehat{j}_{A_i} \in \mathcal{S} \text{ and } \max_j VI_{A_i}(j) > 2VI_{A_i}\big(\widehat{j}_{A_i}\big)\right\}^c\right). \end{aligned} \tag{7.18}$$

For all internal nodes in $\mathcal{A}_1$, the number of nonmuted variables is less than or equal to p. Hence, by the monotonicity of $\eta(\cdot)$ in Assumption 4.6 and Equation 7.17, the first term in Equation 7.18 can be bounded above by

$$\sum_{A_i \in \mathcal{A}_1} C \cdot e^{-\psi_1(n^{\gamma^*-1}) \cdot \psi_2(n^{\gamma^*-1}) \cdot n^{\gamma^* \eta(p)}/K}. \tag{7.19}$$

Note that in $\mathcal{A}_2$, the node sample size is less than n^{γ^*}. Since we choose the splitting point uniformly between the q-th and $(1-q)$-th quintile, to reach a node in $\mathcal{A}_2$, we need to go through a minimal of $-\gamma^* \log_q(n)$ splits. Noticing that this number goes to infinity, and that we mute p_d variables after each split, all variables except the ones in the protected set should be muted in $\mathcal{A}_2$. Hence, the second term in Equation 7.18 can be bounded above by

$$\sum_{A_i \in \mathcal{A}_2} C \cdot e^{-\psi_1(n^{\gamma-1}) \cdot \psi_2(n^{\gamma-1}) \cdot n^{\gamma \eta(p_0)}/K}. \tag{7.20}$$

Noting that $\mathcal{A}_1 \bigcup \mathcal{A}_1 = \mathcal{A}$, and that they contain at most $n^{1-\gamma}$ elements, and combining Equations 7.19 and 7.20, we obtain:

$$P\left(\left\{\widehat{j}_A \in \mathcal{S} \text{ and } \max_j VI_A(j) > 2VI_A\big(\widehat{j}_A\big), \text{ for all } A_i \in \mathcal{A}\right\}^c\right)$$

$$\leq \quad C \cdot n^{1-\gamma} e^{-\left\{\psi_1(n^{\gamma^*-1})\cdot\psi_2(n^{\gamma^*-1})\cdot n^{\gamma^*\eta(p)}+\psi_1(n^{\gamma-1})\cdot\psi_2(n^{\gamma-1})\cdot n^{\gamma\eta(p_0)}\right\}/K},$$

which goes to zero at an exponential rate. Thus the desired result in this step is established.

Step 2: Now we start by decomposing the total variation and bounding it by the variable importance:

$$\begin{aligned} \mathbb{E}[(\widehat{f}-f)^2] &= \int (\widehat{f}-f)^2 d\,\mathbb{P} \\ &= \sum_t \int_{A_t} (\widehat{f}-\bar{f}_{A_t})^2 d\,\mathbb{P} + \sum_t \int_{A_t} (\bar{f}_{A_t}-f)^2 d\,\mathbb{P}, \end{aligned} \tag{7.21}$$

where $\bar{f}_{A_t}$ is the conditional mean of f within terminal node A_t, and where t indexes the terminal node. Noting that each terminal node A_t in $\widehat{f}$ contains $n_{A_t} \geq n^\gamma$ observations, and that the value of $\widehat{f}$ at each terminal node is the average of the Ys, it must therefore have an exponential tail. Hence the first term in Equation (7.21) can be bounded by:

$$\begin{aligned} \sum_t \int_{A_t} (\widehat{f}-\bar{f}_{A_t})^2 d\,\mathbb{P} &\leq \sum_t P(A_t)\cdot(\mathbb{P}_{n_{A_t}}-\mathbb{P}_{A_t})f \\ &= \sum_t P(A_t)\cdot O_p(n_{A_t}^{\frac{1}{2}}) \\ &\leq O_p(n^{-\gamma/2}). \end{aligned} \tag{7.22}$$

The second sum in Equation (7.21) can be further expanded as

$$\begin{aligned} &\sum_t \int_{A_t} (\bar{f}_{A_t}-f)^2 d\,\mathbb{P} \\ &= \sum_t \int_{\mathbf{X}\in A_t} (\bar{f}_{A_t}-f(\mathbf{X}))^2 d\mathbf{X} \\ &= \sum_t \int_{\mathbf{X}\in A_t} \left(\int_{\mathbf{Z}\in A_t} f(\mathbf{Z}) \frac{d\mathbf{Z}}{P(A_t)} - f(\mathbf{X}) \right)^2 d\mathbf{X} \\ &= \sum_t \int_{\mathbf{X}\in A_t} \left(\int_{\mathbf{Z}\in A_t} \big(f(\mathbf{Z})-f(\mathbf{X})\big) \frac{d\mathbf{Z}}{P(A_t)} \right)^2 d\mathbf{X}. \end{aligned} \tag{7.23}$$

The Cauchy-Schwartz inequality now implies that

$$\sum_t \int_{A_t} (\bar{f}_{A_t}-f)^2 d\,\mathbb{P}$$

$$\begin{aligned} &\leq \sum_t \int_{\mathbf{X}\in A_t}\int_{\mathbf{Z}\in A_t} \big(f(\mathbf{Z})-f(\mathbf{X})\big)^2 \frac{d\mathbf{Z}}{P(A_t)} d\mathbf{X} \\ &= \sum_t \int_{\mathbf{X}\in A_t} E\big[\big(f(\mathbf{Z})-f(\mathbf{X})\big)^2 |\, \mathbf{Z}\in A_t\big] d\mathbf{X} \\ &= \sum_t P(A_t)\cdot E\big[\big(f(\mathbf{Z})-f(\mathbf{X})\big)^2 |\, \mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\big]. \end{aligned} \tag{7.24}$$

For each given A_t, due to the independence of $\mathbf{Z}$ and $\mathbf{X}$, the expectation in every summand can be decomposed as

$$\begin{aligned} &E\big[\big(f(\mathbf{Z})-f(\mathbf{X})\big)^2 |\, \mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\big] \\ &= E\big[\big(f(Z^{(1)},...,Z^{(p)})-f(X^{(1)},...,X^{(p)})\big)^2 |\, \mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\big] \\ &= E\Big[\big(f(Z^{(1)},Z^{(2)},...,Z^{(p)})-f(X^{(1)},Z^{(2)},...,Z^{(p)})\big)^2 \\ &\quad +\big(f(X^{(1)},Z^{(2)},Z^{(2)},...,Z^{(p)})-f(X^{(1)},X^{(2)},Z^{(3)},...,Z^{(p)})\big)^2 \\ &\quad +\cdots \\ &\quad +\big(f(X^{(1)},...,X^{(p_1-1)},Z^{(p_1)},...,Z^{(p)})-f(X^{(1)},...,X^{(p)})\big)^2 |\, \mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\Big]. \end{aligned} \tag{7.25}$$

Note that the variables with the labels $p_1+1,...,p$ are in the set $\mathcal{S}^c$ of noise variables. Changing the values of these components will not change the value of f. Hence the last term in the expectation of (7.25) is equal to

$$\big(f(X^{(1)},...,X^{(p_1-1)},Z^{(p_1)},X^{(P_1+1)}...,X^{(p)})-f(X^{(1)},...,X^{(p)})\big)^2.$$

Again, since all the components of $\mathbf{X}$ and $\mathbf{Z}$ are independent, the jth term in the expectation of (7.25) corresponds to the variable importance of the jth variable. Thus we have:

$$\begin{aligned} &E\big[\big(f(\mathbf{Z})-f(\mathbf{X})\big)^2 |\mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\big] \\ &= E\Big[\big(f(Z^{(1)},Z^{(2)},...,Z^{(p)})-f(X^{(1)},Z^{(2)},...,Z^{(p)})\big)^2 |\mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\Big] \\ &\quad +E\Big[\big(f(X^{(1)},Z^{(2)},Z^{(2)},...,Z^{(p)})-f(X^{(1)},X^{(2)},Z^{(3)},...,Z^{(p)})\big)^2 \Big| \mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\Big] \\ &\quad +\cdots \\ &\quad +E\Big[\big(f(X^{(1)},...,X^{(p_1-1)},Z^{(p_1)},...,Z^{(p)})-f(X^{(1)},...,X^{(p)})\big)^2 |\mathbf{Z}\in A_t,\ \mathbf{X}\in A_t\Big] \\ &= \sum_{j=1}^{p_1} VI_{A_t}(j) \end{aligned}$$

$$\leq \quad p_1 \max_j VI_{A_t}(j). \tag{7.26}$$

It remains to show that $\max_j VI_{A_t}(j) \to 0$ as $n \to \infty$. Using Lemma 7.1, we have $\max_j VI_{A_t}(j) = o(n^{-C_1})$ where C_1 depends only on γ, p_1, and q. Moreover, the definition of C_1 shows that it is a strictly decreasing function of p_1. Hence

$$\begin{aligned} & E\big[\big(f(\mathbf{Z}) - f(\mathbf{X})\big)^2 | \mathbf{Z} \in A_t,\ \mathbf{X} \in A_t\big] \\ \leq \quad & C_2 \times O_p(n^{-C_1}). \end{aligned} \tag{7.27}$$

Combining equations (7.21), (7.22) and (7.27), we have

$$\mathbb{E}[(\widehat{f} - f)^2] \quad = \quad O_p(n^{-C_3}), \tag{7.28}$$

where $C_3 = (\min(C_1, \gamma/2))$. Due to the monotonicity of C_1, C_3 is also monotone decreasing in p_1. Noticing that C_3 does not depend on p, the convergence rate of RLT only depends on the choice of γ, q, and the number of strong variables p_1. This concludes the proof. □

LEMMA 7.1. *Let $\mathcal{A}_{nT}$ denote the set of the terminal hypercubes. Then it holds*

$$\max_{A \in \mathcal{A}_{nT}, j \in \mathcal{S}} VI_A(j) = O_p(n^{-C}),$$

where C is a constant depending only on γ, p_1, and q.

PROOF OF LEMMA 7.1. For any terminal hypercube $A \in \mathcal{A}_{nT}$, let $A_1 \to A_2 \to \ldots \to A_N = A$ be the constructed chain of the nodes leading to A, where A_{k+1} is the daughter node of A_k. Since at each node, the splitting point is chosen uniformly between the $100q$ and $100(1-q)$ quantiles of the current range of the splitting variable for some $q \in (0, \frac{1}{2})$, and since the terminal node is the last node having $\geq n^\gamma$ observations, it is easy to see that $-\gamma \log_q(n) \leq N \leq -\gamma \log_{(1-q)}(n)$. Let $j_k = \text{argmax}_{j \in \mathcal{S}} \widehat{VI}_{A_k}(j)$ be the index of the variable selected for splitting at node A_k and, moreover, define $m_j = \sum_{k=1}^N I(j_k = j)$, the number of times the jth variable is used for splitting. Let $N_j = \max\{k :, k = 1, ..., N, j_k = j\}$, the index of the last node split with the jth variable.

Before presenting the main proof, we state two simple properties:
Property 1. For $j \in \mathcal{S}$, $VI_{A_{N_j}}(j) \leq c_1(1-q)^{m_j}$. This is because after node A_{N_j}, the interval of the jth variable has been split m_j times so its length is at most $(1-q)^{m_j - 1}$. Therefore, according to the proof of Theorem 4.7,

$VI_{A_{N_j}}(j) \le c_1(1-q)^{2m_j}$.

Property 2. For $k = 1, ..., N-1$ and any $j \in \mathcal{S}$, $V_{A_{k+1}}(j) \le 2VI_{A_k}(j_k)/q^2$. That is, the importance of any variable in the daughter node is no larger than the importance of the selected variable at the current node by a factor of $2/q^2$. This follows from Theorem 4.7 (b): $2VI_{A_k}(j_k) \ge \max_j VI_{A_k}(j)$. On other hand, for any $j \in \mathcal{S}$, since $A_{k+1} \subset A_k$ and $|A_{k+1}| \ge |A_k|/q$, we have

$$\begin{aligned} VI_{A_k}(j) &= \frac{E\Big[\big(f(X^{(-j)}, X^{(j)}) - f(X^{(-j)}, \tilde{X}^{(j)})\big)^2 I(X \in A_k, \tilde{X} \in A_k)\Big]}{\sigma^2 P(X \in A_k)} \\ &\ge \frac{E\Big[\big(f(X^{(-j)}, X^{(j)}) - f(X^{(-j)}, \tilde{X}^{(j)})\big)^2 I(X \in A_{k+1}, \tilde{X} \in A_{k+1})\Big]/q}{\sigma^2 P(X \in A_{k+1}) q} \\ &= VI_{A_{k+1}}(j)/q^2. \end{aligned}$$

Thus, $V_{A_{k+1}}(j) \le VI_{A_k}(j)/q^2 \le 2VI_{A_k}(j_k)/q^2$. With these two properties, we now proceed to prove the lemma. First, we define the following sequence:

$$N > \frac{N}{(rp_1)^1} > \cdots > \frac{N}{(rp_1)^{p_1}} > 0, \tag{7.29}$$

where r is a constant satisfying $r > 1$ and $2(1-q)^{2r}/q^2 = c \le 1$. Since $0 < q < 1/2$, r can always be properly chosen. Correspondingly, we obtain intervals $W_k = [N/(rp_1)^k, N/(rp_1)^{k-1})$ for $k = 1, ..., p_1$ and $W_{p_1+1} = [0, N/(rp_1)^{p_1})$. Recall the definition of m_j, the number of times the jth variable is selected for splitting. Since $\sum_{k=1}^{p_1} m_j = N$, there must be at least one j such that $m_j \ge N/(rp_1)$ and $m_j \in W_1$. Furthermore, since there are (p_1+1) intervals, there exists an integer $p_1 + 1 \ge k_0 \ge 2$ such that $m_j \notin W_{k_0}$ for any $j = 1, ..., p_1$. Hence, we can define two sets:

$$\mathcal{S}_1 = \{j : m_j \ge N/(rp_1)^{k_0-1}\},$$

and

$$\mathcal{S}_2 = \{j : m_j < N/(rp_1)^{k_0}\},$$

so that $\mathcal{S}_1 \ne \emptyset$ and $\mathcal{S}_1 \cup \mathcal{S}_2 = \{1, ..., p_1\}$.

Let j^* be the variable in $\mathcal{S}_1$ and split last among all the variables in $\mathcal{S}_1$ and let N^* be the node index where this variable is split last. In other words, the variables selected in the nodes A_k for $k > N^*$ are all from $\mathcal{S}_2$. Then using Property 1, we have $VI_{A_{N^*}}(j^*) \le c_1(1-q)^{2m_{j^*}}$. Using the fact that $j^* \in \mathcal{S}_1$, we obtain

$$V_{A_{N^*}}(j^*) \le c_2(1-q)^{2N/(rp_1)^{k_0-1}}.$$

Since all splitting variables after node A_{N^*} are from $\mathcal{S}_2$, and the number of the distinct variables is at most (p_1-1), and the number of possible splits after $A_{N^*} = N - N^*$, is no larger than $(p_1-1)N/(rp_1)^{k_0}$. Hence we conclude: (a) if $N^* = N$, then

$$\begin{aligned} VI_A(j) &= VI_{A_N}(j) \le 2VI_{A_N}(j_N) = 2VI_{A_{N^*}}(j^*) \\ &\le 2c_1(1-q)^{2N/(rp_1)^{k_0-1}} 2c_1 \le (1-q)^{2N/(rp_1)^{p_1}}. \end{aligned}$$

(b) if $N^* < N$, then according to Property 2,

$$VI_A(j) = VI_{A_N}(j) \le (\frac{2}{q^2})^{N-N^*} VI_{A_N^*}(j^*) \le (\frac{2}{q^2})^{(p_1-1)N/(rp_1)^{k_0}} VI_{A_N^*}(j^*).$$

Thus,

$$\begin{aligned} VI_A(j) &\le \frac{2c_3}{(1-q)^2q^2}\left(\frac{2(1-q)^{2r}}{q^2}\right)^{(p_1-1)N/(rp_1)^{k_0}} (1-q)^{2rN/(rp_1)^{k_0}} \\ &\le c_4(1-q)^{2rN/(rp_1)^{k_0}} \\ &\le c_4(1-q)^{2rN/(rp_1)^{p_1+1}}, \end{aligned} \tag{7.30}$$

where c_4 is a constant depending on p_1 and q, and where we used the fact that $2(1-q)^{2r}/q^2 < 1$.

Finally, since $-\gamma \log_q(n) \le N \le -\gamma \log_{(1-q)}(n)$, we obtain

$$\max_{j\in\mathcal{S}} VI_A(j) \le c_5(1-q)^{-2r\gamma \log_q(n)/(rp_1)^{p_1+1}},$$

where c_5 is a constant depending only on p_1, q and R. The lemma holds. □

References.

Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computing*, 9(7):1545C1588, 1997.

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

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

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

L. Breiman. Some infinity theory for predictor ensembles. Technical Report 577, Department of Statistics, University of California, Berkeley, 2000.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman. Consistency for a simple model of random forests. Technical Report 670, Department of Statistics, University of California, Berkeley, 2004.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bart: Bayesian additive regression trees. *Ann. Appl. Stat*, 4(1):266–298, 2010.

A. Cutler and G. Zhao. Pert – perfect random tree ensembles. *Computing Science and Statistics*, 33:??, 2001.

T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40:139–157, 2000.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statistics*, 29:1189–1232, 2001.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101:578–590, 2006.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

S. A. van de Geer and J. C. Lederer. The bernstein-orlicz norm and deviation inequalities. *preprint*, 2011.

A. van der Vaart and J. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer, 1996.

R. Zhu and M. R. Kosorok. Recursively imputed survival trees. *Journal of the American Statistical Association*, 107:331–340, 2012.

Ruoqing Zhu, Donglin Zeng, and Michael R. Kosorok
Department of Biostatistics
The University of North Carolina at Chapel Hill
Chapel Hill, NC 27599, U.S.A.
E-mail: rzhu@live.unc.edu
dzeng@email.unc.edu
kosorok@unc.edu