Regression Trees Know Calculus

Nathan Wycoff
Department of Mathematics and Statistics
University of Massachusetts
Amherst, MA 01003
nwycoff@umass.edu

Abstract

Regression trees have emerged as a preeminent tool for solving real-world regression problems due to their ability to deal with nonlinearities, interaction effects and sharp discontinuities. In this article, we rather study regression trees applied to well-behaved, differentiable functions, and determine the relationship between node parameters and the local gradient of the function being approximated. We find a simple estimate of the gradient which can be efficiently computed using quantities exposed by popular tree learning libraries. This allows tools developed in the context of differentiable algorithms, like neural nets and Gaussian processes, to be deployed to tree-based models. To demonstrate this, we study measures of model sensitivity defined in terms of integro-differential quantities and demonstrate how to compute them for regression trees using the proposed gradient estimates. Quantitative and qualitative numerical experiments reveal the capability of gradients estimated by regression trees to improve predictive analysis, solve tasks in uncertainty quantification, and provide interpretation of model behavior.

1 Introduction

Tree-based methods, such as regression trees, are a workhorse of the contemporary data scientist. Their ease of use, computational efficiency and predictive capability without the need for extensive feature engineering makes them popular with practitioners. The most widely used version of regression trees approximate with greedily constructed, piecewise-constant functions than can handle data which exhibit discontinuities or divergent behavior in various parts of the feature-space. Perhaps because of their capability to tackle pathological problems, it seems that some of their properties in approximating well-behaved functions may have gone unnoticed.

In this article, we will study the approximation of a well-behaved differentiable function f on the unit cube in dimension P with a piecewise constant regression tree. In particular, we will investigate a means of approximating ∇f using only information contained in the tree structure computable in a single pass through the tree. We find a simple and easily computable quantity analogous to a finite difference and use it to rapidly form estimates of integro-differential quantities. Previously, gradient estimation in regression trees has been studied in the context where the leaves have differentiable models, and the gradients of these models are used to estimate the gradient (e.g. Chaudhuri et al. (1995); Loh (2011)). However, the constant-leaf tree remains prevalent in practice, and the purpose of this article is rather to examine the *implicit* gradient estimation that occurs within these constant-leaf trees where, formally, the gradient of the tree is almost-everywhere zero.

With a gradient estimator in hand, we unlock for tree-based models the stable of existing gradient-based methods for variable interpretation and dimension reduction developed in other areas. Among many possibilities, we will study in particular the Active Subspace Method (Constantine, 2015), a global dimension reduction technique from the Uncertainty Quantification literature, and the

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

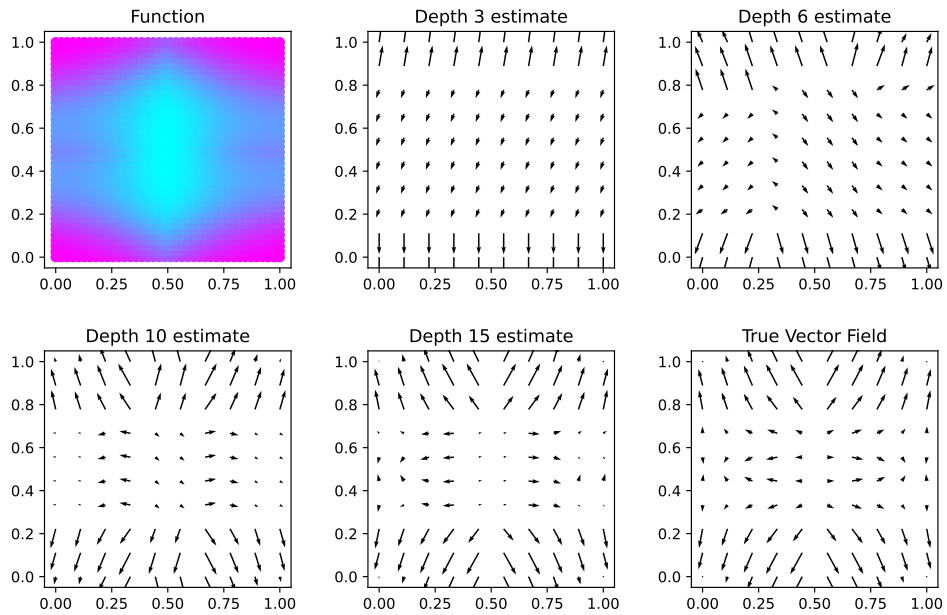

Figure 1: **Illustration of Gradient Estimates**. The top left gives a target function, and the bottom right gives its gradient vector field. Shown in between are estimates of the gradient extracted from a regression tree fit to data from the function converging to the true vector field.

Integrated Gradient Method (Sundararajan et al., 2017), a local model interpretation technique from the neural network literature.

Active Subspaces provide for linear dimension reduction, which is already commonly used in the setting of regressions trees, such as when using random projection or PCA for rotated trees (Breiman, 2001; Rodriguez et al., 2006). In contrast to these methods, however, active subspaces consists of supervised linear dimension reduction which takes the relationship between features and response into account. Already, supervised linear dimension reduction has been proposed for use with tree based methods where, e.g. a kernel method is used to perform the dimension reduction which is then applied to a tree-based method (Shan et al., 2015). But here, we show how to actually use the tree itself to perform a linear sensitivity analysis, rather than relying on a helper model to do this. This is essential if the analyst is interested in *model*-interpretation (as contrasted with *data*-interpretation (Chen et al., 2020)), and to the best of our knowledge the application of the Active Subspace method, enabled by our novel gradient estimates, is the first such linear sensitivity metric for trees. And while we've so far discussed what active subspaces can do for regression trees, we don't think this new relationship will be one-sided. Our numerical experiments show that regression trees can serve as scalable estimators of the active subspace, favorable to existing methods in certain circumstances. We hope this can highlight the potential for regression trees in the gradient-based UQ space.

We view the major contributions of our article as follows:

1. We study a simple algorithm to extract gradient and integrated gradient information from regression trees.
2. We show how to use this to port gradient-based interpretability techniques from other fields to benefit interpretability of regression trees.
3. We find that in certain circumstances gradient-enabled regression trees can produce better estimates of active subspaces than existing UQ methods.

We begin in Section 2 by discussing pertinent background on regression trees and gradient-based interpretability. Our main methodological contributions are given in Section 3 and developed theoretically in Section 4. Subsequently, we illustrate these procedures on numerical examples in Section 5 before offering conclusions and future research directions in Section 6.

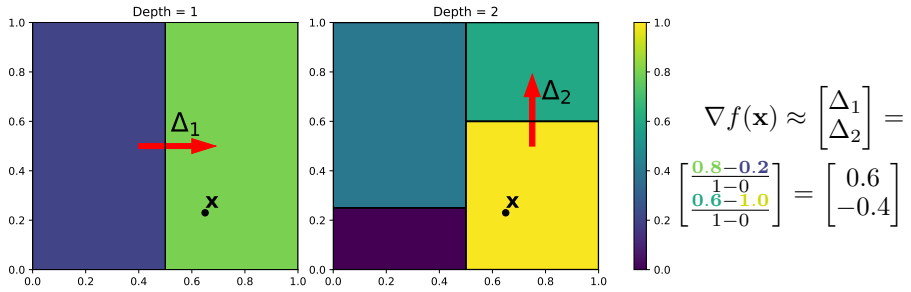

Figure 2: We can extract finite difference gradient approximations from a regression tree by comparing values of adjacent nodes in splits.

2 Background

While small decision trees are easily interpretable, they become significantly less so as they grow in size or if they are aggregated via Random Forests (Breiman, 2001, 2002) or gradient boosting (Friedman, 2001). Consequently, developing methods for increasing the interpretability of trees is of great importance (Louppe, 2014).

2.1 Variable Interpretation and Selection in Regression Trees

Several methods for variable importance were suggested alongside some of the original tree-based methods (Breiman, 2017, 2001, 2002), the two most salient for our study being the Mean Decrease in Impurity (called MDI or TreeWeight Kazemitabar et al. (2017)) and the Feature Permutation method. MDI assigns importance to variables in proportion to the mean reduction in cost when splitting along a given variable, an intuitively reasonable approach. However, MDI as originally proposed was sensitive to properties of the feature variables as well as to the depth within the tree that a split occurred, leading to "debiased" variants (Sandri and Zuccolotto, 2008; Li et al., 2019). Kazemitabar et al. (2017) as well as Klusowski and Tian (2020) studied theoretical properties of a simplified version of this metric using only the first split ("stumps"). On the other hand Feature Permutation is based on measuring decrease in performance when shuffling a given feature, and though the method does have some desirable properties (Ishwaran, 2007), it has come under criticism for undesirable behavior in the context of correlated predictors (Hooker et al., 2021). Empirical studies have found issues in both of these methods (Strobl et al., 2007, 2008). Against this backdrop of negative results on the behavior of these initially suggested methods came contributions in general-purpose machine learning interpretability methods. SHAP values (Lundberg and Lee, 2017) have been proposed as a general-purpose tool for understanding machine learning models. However, they would prove especially popular for understanding tree-based methods where efficient algorithms have been proposed for estimating these otherwise combinatorially difficult quantities (Lundberg et al., 2019; Karczmarz et al., 2022).

2.2 Gradient-based Model Interpretation

The gradient of a function tells us how its output is related to its input over short distances, and is a natural candidate to help explain the behavior of a model. Hechtlinger (2016) suggested to simply look at the gradient of a model evaluated at a particular observation to gain local understanding. Another approach is the Integrated Gradient (Sundararajan et al., 2017), introduced in the context of neural networks, which involves privileging some setting of input features which is called the *reference point*, which we'll denote $\mathbf{x}^*$. Subsequently, in order to explain a prediction at a given point $\mathbf{x}$, we integrate the gradient along the path from $\mathbf{x}^*$ to $\mathbf{x}$ and multiply elementwise against that difference; that is, denoting the output of the neural network as f:

$$IG(\mathbf{x}) := (\mathbf{x} - \mathbf{x}^*) \odot \int_{\alpha=0}^{1} \nabla f(\alpha\mathbf{x} + (1-\alpha)\mathbf{x}^*) d\alpha \,. \tag{1}$$

This theoretically appealing approach has also seen practical success in applications including medicine (Sayres et al., 2019) and chemistry (McCloskey et al., 2019).

One could also integrate the gradient over a larger part of the space to perform a more global sensitivity analysis. This is the idea behind the Active Subspace Method (Constantine, 2015). In our context, this involves defining the Active Subspace Matrix as:

$$\mathbf{C}_\mu^f := \int_{[0,1]^P} \nabla f(\mathbf{x}) \nabla f(\mathbf{x})^\top d\mu(\mathbf{x}) \,, \tag{2}$$

where μ is a measure. Eigenanalysis on $\mathbf{C}_\mu^f$ reveals linear combinations of features which are important, similar to PCA. Various techniques have been developed to estimate active subspaces in the presence only of input-output data. A Polynomial Ridge Approximation (PRA) procedure (Hokanson and Constantine, 2018) is available, as is a closed form estimate using Gaussian processes Wycoff et al. (GP; 2021), and also an approach based on neural networks called the Deep Active Subspace Method (DASM; Tripathy and Bilionis, 2019; Edeling, 2023) has been proposed.

2.3 Variable Importance versus Sensitivity Analysis

Though originating in different academic communities, the ideas of Variable Importance (from the Machine Learning community) and Sensitivity Analysis (from the Uncertainty Quantification community) are closely related. See Antoniadis et al. (2021) for a recent review of using regression trees for (non-gradient-based) sensitivity analysis. Vannucci et al. (2024) explicitly port ideas from the sensitivity analysis literature to enhance variable importance in tree-based methods. Elie-Dit-Cosaque and Maume-Deschamps (2024) use regression trees to estimate a form of sensitivity analysis based on quantiles.

3 Integro-Differentiation of Regression Trees

In this section we propose an easily computed estimator of derivatives and integrals of derivatives for regression trees. While the underlying idea is simple, discussing it rigorously requires a fair amount of notation; Figure 3 illustrates our notation on a simple regression tree. Say we fit a regression tree of depth K to data $(\mathbf{x}_n, y_n = f(\mathbf{x}_n) + \epsilon_n)$ with $\mathbf{x}_n \in [0,1]^P, y_n \in \mathbb{R}$ for $n \in \{1, \ldots, N\}$, and ϵ_n mean zero with finite variance. Purely for simplicity, we will assume that the tree has leaf nodes only at depth K, i.e. there is no path through the tree that ends before depth K. Associate with every node from all depths an integer index i such that the root node is labeled 0 and if node i is deeper in the tree than node j, then $i > j$. That is, the indices at depth 1 are given by $\{1, 2\}$, at depth 2 by $\{3, 4, 5, 6\}$, and generally at depth k the indices are given by $\mathcal{N}_k = \{2^{k-1} + 1, \ldots, 2^k\}$. For any depth k, we define the function $B^k : [0,1]^P \to \mathcal{N}_k$ where $B^k(\mathbf{x})$ gives the index of the node at depth k which contains $\mathbf{x}$. We denote the variable along which the i^{th} node splits by $\sigma_i \in \{1, \ldots, P\}$ and the threshold at which that split occurs as τ^i. Those points in node i for which $x_{\sigma_i} \leq \tau^i$ are mapped to the left child node of i, denoted c_l^i, while those for which $x_{\sigma_i} > \tau^i$ are mapped to the right child node, denoted c_r^i. For each node i with child nodes, we denote the mean of the response variable in the left child node by $\hat{\mu}_l^i$ and in the right child node by $\hat{\mu}_r^i$. That is, $\hat{\mu}_l^i$ gives the average of y_n for all $n \in \{1, \ldots N\}$ such that $B^{D_i+1}(\mathbf{x}_n) = c_l^i$ where D_i is the depth of node i (and so $D_i + 1$ is the depth of its child nodes).

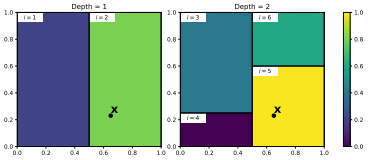

Figure 3: **Illustration of Notation.** In the same example tree as Figure 2 with a depth $K = 2$, examples of our notation is as follows: the indices of nodes at each depth are $\mathcal{D}_1 = \{1, 2\}$ and $\mathcal{D}_2 = \{3, 4, 5, 6\}$; the children of node 2 are $c_l^2 = 5$ and $c_r^2 = 6$; conversely the parent of node 5 is given by $\rho_5 = 2$; the bounds of node 5 are $\mathbf{l}^5 = [0.5, 0]$ and $\mathbf{u}^5 = [1, 0.6]$; the value of intermediate node 2 is $v_2 = 0.8$ and the value of leaf node 6 is 0.6; since the "root node" 0 is split along the x-axis, $\sigma_0 = 1$ and since nodes 1 and 2 are split along the y-axis, $\sigma_1 = \sigma_2 = 2$; since the point $\mathbf{x}$ lies within the nodes 2 and 5 at depths 1 and 2 respectively, we have that $B^1(\mathbf{x}) = 2$ and $B^2(\mathbf{x}) = 5$.

3.1 Differentiation

The difference between $\hat{\mu}_l^i$ and $\hat{\mu}_r^i$, which contain the mean of the data on either side of the threshold for node i, contains information about the σ_i^{th} component of the gradient in that area. By dividing this

difference proportional to the size of the node along dimension σ_i, we form a quantity similar to a finite difference. Let $[l_p^i, u_p^i]$ denote the extent of node i along dimension p for $p \in \{1, \ldots, P\}$. Then define the following quantity:

$$\gamma_i := \frac{2(\hat{\mu}_r^i - \hat{\mu}_l^i)}{u_{\sigma_i}^i - l_{\sigma_i}^i} \approx \frac{\partial f}{\partial x_{\sigma_i}}(\mathbf{x}) \; \forall \mathbf{x} \in [\mathbf{l}^i, \mathbf{u}^i]\,. \tag{3}$$

We have used the notation $\mathbf{l}^i = [l_1^i, \ldots, l_P^i]$ to denote the vector of lower bounds of the extent of node i and similar for $\mathbf{u}^i$, and by $[\mathbf{l}^i, \mathbf{u}^i]$ we denote the set of points lying within the bounds of the i^{th} node. The ratio γ_i contains information about the partial derivative of f with respect to x_{σ_i} (which is, again, the variable that node i splits along) inside of $[\mathbf{l}^i, \mathbf{u}^i]$. The idea that $\gamma_i \approx \frac{\partial f(\mathbf{x})}{\partial x_{\sigma_i}}\big|_i$ for any $\mathbf{x}$ in node i only makes sense if the gradient does not vary much within it. We expect this to be the case only for very small nodes, or in other words, very deep trees, estimating functions with gradients which vary smoothly; we make this precise in the next section.

At a particular node, we can only form estimates of a single partial derivative. However, if our tree is sufficiently deep, we can combine estimates of partial derivatives from nodes at multiple depths to form an estimate of the entire gradient. Recall that $\mathcal{N}_k = \{2^{k-1} + 1, \ldots, 2^k\}$ contains the index of all nodes of depth k and let ρ_i denote the index of the parent of node i (which will be a member of $\mathcal{N}_{k-1}$). Then Algorithm 1 shows how to aggregate the partial derivative estimates into a gradient estimate.

Algorithm 1 Tree-Based Gradient Estimation

$\mathbf{G}^i \leftarrow \mathbf{0} \in \mathbb{R}^P$ for all i.
for $k \in \{1, \ldots, K\}$ **do**
  for $i \in \mathcal{N}_k$ **do**
    $\mathbf{G}^i \leftarrow \mathbf{G}^{\rho_i}$ {Get parent's estimate}
    $\mathbf{G}^i[\sigma_i] \leftarrow \frac{2(\hat{\mu}_r^i - \hat{\mu}_l^i)}{u_{\sigma_i}^i - l_{\sigma_i}^i}$ {Update along split direction}
  end for
end for

We thus have associated each node i with an estimator of the gradient, though if we have not split along a variable p in the tree upstream of i, the estimate of that component will simply be 0. This is actually somewhat reasonable, as the tree will split along variables with small gradient components less frequently.

Of course, nothing would stop us from comparing values of means that are adjacent in the input domain but not adjacent in the tree structure (for example, nodes 3 and 6 of Figure 3). However, the advantage of considering only nodes adjacent in the tree is that this Tree-Based Gradient Estimator (TBGE) can be computed efficiently merely by traversing the tree. Furthermore, if we needed only an estimate of a gradient at a particular point $\mathbf{x}$, we would need only to traverse the regression tree in the standard manner, visiting only nodes upstream of the leaf node $\mathbf{x}$. To be explicit, if the tree is of depth K, recall that $B^K(\mathbf{x})$ gives the index of the node of the greatest depth which contains $\mathbf{x}$. We use as our estimate of $\nabla f(\mathbf{x})$ the TBGE associated that node; that is, $\tilde{\nabla} f(\mathbf{x}) := \mathbf{G}^{B^K(\mathbf{x})}$.

3.2 Integration

We next develop estimators for quantities of the form:

$$\mathcal{I}(f) = \int h(\nabla f(\mathbf{x})) d\mu(\mathbf{x})\,. \tag{4}$$

Here, h is some integrable $\mathbb{R}^P \to \mathcal{H}$ function and μ is a measure on $\mathbb{R}^P$. The Integrated Gradient at point $\mathbf{z}$ with reference point $\mathbf{z}^*$ may be written in this form by choosing $h(\mathbf{a}) = (\mathbf{z} - \mathbf{z}^*) \odot \mathbf{a}$, $\mathcal{H} = \mathbb{R}^P$, and μ as the degenerate uniform measure on the line segment between $\mathbf{z}$ and $\mathbf{z}^*$. Similarly, the Active Subspace matrix can also be seen to belong to this class by setting $h(\mathbf{a}) = \mathbf{a}\mathbf{a}^\top$, $\mathcal{H} = \mathbb{R}^{P \times P}$ and arbitrary μ.

We will consider two classes of estimators for such quantities. The first is a Monte Carlo Estimator (MCE) which is appropriate whenever μ is a probability measure which it is possible to sample

from. The second is a Partition-Based Estimator (PBE) which is suitable whenever we can compute $\mu([\mathbf{a}, \mathbf{b}])$ for arbitrary $\mathbf{a}, \mathbf{b}$.

Start with the MCE. We fix some Monte Carlo sample size M and form the standard Monte Carlo approximation, then plug in the TBGE where the gradient appears:

$$\mathcal{I}(f) \approx \frac{1}{M} \sum_{\mathbf{x}_m \sim \mu} h(\nabla f(\mathbf{x}_m)) \approx \frac{1}{M} \sum_{\mathbf{x}_m \sim \mu} h(\tilde{\nabla} f(\mathbf{x}_m)) \,. \tag{5}$$

We denote this estimator by $\hat{\mathcal{I}}_{MC}(f)$. In particular, we can form a Tree-Based Integrated Gradient (TBIG) as follows by sampling random uniform variables u_m on $[0, 1]$ and then forming:

$$\hat{IG}(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*) \odot \frac{1}{M} \sum_{m=1}^{M} \tilde{\nabla} f(u_m \mathbf{x} + (1 - u_m)\mathbf{x}^*) \,. \tag{6}$$

Next, we consider the PBE. This involves simply computing a weighted average of the action of h on the TBGE on each node at the penultimate depth $K - 1$, weighted by the measure of that node. For simplicity, we assume that $\mu([0, 1]^P) = 1$. In notation, we define the estimator as

$$\hat{\mathcal{I}}_{PE}(f) = \sum_{i \in \mathcal{N}_{K-1}} h(\mathbf{G}^i) \mu([\mathbf{l}^i, \mathbf{u}^i]) \,. \tag{7}$$

We can use this to form Tree-Based Active Subspace (TBAS) for f as follows:

$$\hat{C}_\mu^f = \sum_{i \in \mathcal{N}_{K-1}} \mathbf{G}^i \mathbf{G}^{i\top} \mu([\mathbf{l}^i, \mathbf{u}^i]) \,. \tag{8}$$

Which estimator is preferred in practice will depend on which properties of μ are computationally tractable. The ability to sample from μ makes the MCE tractable while the ability to compute the measure of an arbitrary rectangle makes the PBE tractable. If both are available, the PBE will be favorable as it does not have Monte Carlo error.

4 Asymptotic Analysis

We'll now make rigorous the intuition developed in the previous section with a basic asymptotic analysis of the TBGE under simplifying assumptions. In particular, we assume that the split decisions are made on the basis of the feature variables only without reference to the outcome variables. Though this does not accommodate most algorithms used in practice for supervised learning, such as the the CART algorithm, these assumptions nevertheless are sufficient to develop some intuition as to qualitative properties of the approximation. Additionally, similar simplifying assumptions have often been used in recent articles (e.g. Gao et al., 2022; Ma et al., 2023; Cai et al., 2023; Wen and Hang, 2022; Klusowski, 2021).

Theorem 4.1. *Let $S(N)$ denote the number of splits in the tree as a function of the sample size. Let $\frac{\tilde{\partial}^N f(\mathbf{x})}{\tilde{\partial} x_p}$ denote the TBGE at point $\mathbf{x} \in [0, 1]^P$ with sample size N. Under Assumptions 0, 1, and 2 of the Appendix, we have that:*

$$\frac{\tilde{\partial}^N f(\mathbf{x})}{\tilde{\partial} x_p} = \frac{\partial f(\mathbf{x})}{\partial x_p} + O_P\left(\frac{2^{\frac{P+2}{2P} S(N)}}{\sqrt{N}} + P 2^{-\frac{S(N)}{P}} \right) .$$

In order for that error term to vanish, we need to choose $S(N)$ growing even slower than $\log N$, such as $P \log \log N$ (see Appendix for more discussion). This choice yields an error rate of the gradient estimates of $O_P(P(\log N)^{-1})$, which is slow relative to typical estimation rates. This suggests that this gradient estimator may be most useful in situations where a coarse estimate is acceptable. The reason that this rate needs to be so slow is that on the one hand, the number of datapoints in each cell needs to diverge in order for the statistical error to vanish (given by the first term in the big O expression). But on the other hand, the Taylor series approximation error in the finite difference analog requires for the size of the cells to vanish (given by the second term), which implies an exploding number of cells. These factors combine to require a large number of data to achieve

small errors. However, they leave open the possibility of leveraging the computational efficiency of regression trees to quickly create approximate gradient estimates in large samples, as our numerical experiments will show.

In the remainder of this section, we will establish consistency of estimates for integro-differential quantities, which will require only consistency of the TBGE, which we have just established. We begin with the PBE; the remaining results are in the Appendix.

Theorem 4.2. *Let $h : \mathbb{R}^P \to \mathbb{R}^P$ be bounded in the sense that $h(\mathbf{z}) \leq C_h \|\mathbf{z}\|^{a_h}$ for $a_h > 0$. Under Assumptions 0, 1 and 2 of the appendix, the Partition-Based estimator converges to the true integro-differential quantity as the sample size diverges. That is,*

$$\lim_{N \to \infty} \hat{\mathcal{I}}_{PB}(f) = \mathcal{I}(f)\,.$$

where $\mathcal{I}(f)$ is the integro-differential quantity given in Equation 4.

Theorem A.1 of the Appendix gives a similar result for the MCE. We also explicitly state there the implications of these results for Active Subspace and Integrated Gradient estimation, namely the consistency of the estimators $\hat{C}_\mu^f$ and $\hat{IG}$ proposed in Section 3.

We have thus established consistency of gradient-based model interpretation quantities for regression trees. Next, we will empirically study their behavior in finite samples via a battery of numerical experiments.

5 Numerical Experiments

We now study how the proposed gradient estimator might be profitably exploited in practice. We begin with a qualitative study showing how a tree-based integrated gradient (TBIG) can facilitate local model interpretation. Then come three quantitative studies, first investigating the potential of a Tree-Based Active Subspace (TBAS) to improve prediction accuracy of a downstream tree via a rotation of the space. Subsequently, we evaluate the capacity of regression trees to estimate the Active Subspace in low and high dimension. Finally, we end with another qualitative study demonstrating how a TBAS can provide data visualization.

5.1 Integrated Gradient for Tree-Based Methods

Figure 4: **Integrated Gradient for Trees.** Each pair of panels gives a training example from MNIST. The second pair in the image superimposes the IG values onto the example. Redder means more strongly suggesting correct class membership.

We now consider a random forest classifier fit to two subsets of the MNIST dataset, one contrasting zeros against eights and the other fours against sevens. While classification was not the focus of this article, we present some preliminary numerical experiments in Appendix C.5. We use the estimator $\hat{IG}$ of Equation 6 with $M = 500$ random points along a given path. Figure 4 gives the results of this analysis. We see that when comparing eights against zeros, the intersection point of the eight is most important, while the left and right sides of the zero are. This makes sense as they are the parts of each digit which are distinct. By contrast, the shared upper and lower arcs are not highlighted. In the four versus seven case, the location of the horizontal bar seems to be most influential in determining one versus the other. This suggests that the model is comparing the relative position of the bar to the rest of the digit in making its classifications.

5.2 Active-subspace Rotated Trees for Predictive Analysis

This section evaluates the ability of TBAS to improve the accuracy of a downstream predictive analysis. Given some mapping $\mathbf{L}$, we refer to postmultiplication of the feature matrix $\mathbf{X}$ to form a new

Dataset:	bike	concrete	gas	grid	keggu	kin40k	obesity	supercond
Regression Tree (Depth 4)								
TBAS	**0.635**	**0.47**	**0.578**	**0.688**	**0.194**	**0.856**	**0.128**	**0.511**
Id	**0.655**	0.537	0.595	0.773	0.35	0.963	0.226	**0.514**
PCA	**0.655**	0.543	**0.591**	0.773	0.349	0.964	0.226	**0.513**
Rand	0.656	0.524	0.593	0.752	0.349	0.95	0.226	**0.513**
Regression Tree (Depth 8)								
TBAS	**0.402**	**0.35**	**0.429**	**0.521**	**0.078**	**0.586**	**0.094**	**0.392**
Id	**0.405**	0.403	**0.432**	**0.522**	0.121	0.862	**0.103**	**0.395**
PCA	**0.407**	**0.395**	**0.423**	**0.524**	0.122	0.872	**0.107**	**0.392**
Rand	**0.411**	**0.391**	**0.435**	0.55	0.124	0.836	**0.109**	**0.397**
Random Forest (Depth 4)								
TBAS	**0.609**	**0.406**	**0.559**	**0.602**	**0.161**	**0.802**	**0.123**	**0.479**
Id	0.65	0.462	**0.574**	0.659	0.344	0.954	0.173	**0.485**
PCA	0.649	0.462	**0.567**	0.654	0.344	0.954	0.174	**0.486**
Rand	0.646	0.458	**0.57**	0.66	0.33	0.938	0.173	**0.486**

Table 1: **Predictive Impact of Various Transformations on Selected Datasets**. Numbers give 100-fold RMSE; bold indicates confidence interval overlaps with the lowest confidence interval.

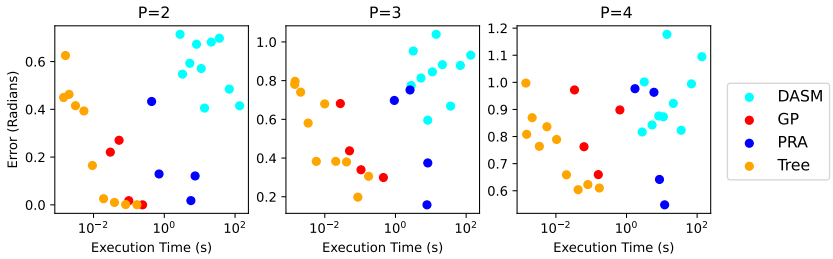

Figure 5: **Active Subspace Estimation in Low Dimension.** Execution time (x-axis) and Subspace Estimation Error (y-axis) for the four methods, lower is better.

feature matrix $\mathbf{Z} := \mathbf{XL}$ as a rotation. In standard tree-based methods, non-diagonal rotations can have an impact on predictive performance because the axes along which splits are made have been changed. In order to quantitatively assess the utility of TBAS, we compare the prediction error of regression trees and random forests on eight benchmark datasets fit on data augmented by a rotation of the space. A TBAS rotation is conducted by choosing $\mathbf{L}$ to be a matrix square root of the active subspace matrix estimate, following Wycoff et al. (2022). We compare TBAS rotations to the those made by drawing orthogonal directions uniformly over the Grassmannian (Rand; similar to Breiman (2001)) as well as those formed from a Principal Components Analysis (PCA; similar to Rodriguez et al. (2006)) and no rotation at all (denoted Id). Table 1 presents the results of this analysis. TBAS does at least as well as the other methods, and sometimes offers significant improvement (such as on the Kin40k and Keggu datasets).

5.3 Computationally Efficient Active Subspace Estimation in Low Dimension

In this section we demonstrate the capability of TBAS to offer extremely fast, gradient-free estimation of the active subspace in low dimension when significant data are available. We compare to the three popular methods for gradient-free active subspace estimation introduced in Section 2.2, based on a Gaussian Process (GP), Ridge Polynomial (PRA), and Neural Network (DASM). Each of these methods was tasked to estimate a one dimensional active subspace in dimensions 2, 3 and 4, using a logarithmically spaced grid of sample sizes ranging from 10 to 10,000 on a toy algebraic function. We found that the GP and PRA methods worked efficiently when sample sizes were small, but

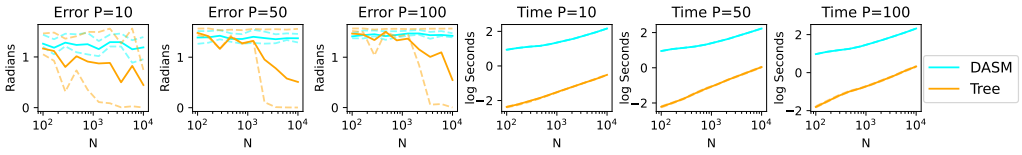

Figure 6: **Sparse Active Subspace Estimation in High Dimension.** Lower is better.

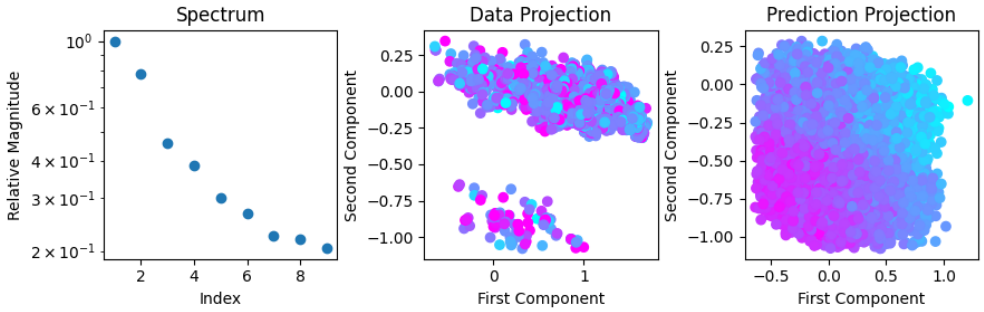

Figure 7: **Projection with TBAS.** *Left:* Spectrum of active subspace matrix suggests a 2D active subspace. *Middle:* Projection of data onto active subspace. *Right:* Projection of predictive surface.

computation time grew quickly, such that we were only able to run these methods for samples of size 150 or less. Figure 5 shows the active subspace error against the execution time. We see that the tree-based estimator forms the majority of the Pareto front in all three cases.

5.4 Estimating Sparse Active Subspaces in High Dimension

Imposing sparsity in the coefficients of linear dimension reduction can improve interpretability (Zou et al., 2006). Because of the manner that tree-based methods produce gradient estimates, we conjecture that TBAS has built-in inductive bias to favor entry-wise active subspaces. In this section, we will investigate this possibility by comparing TBAS against the DASM in estimating a sparse active subspace in dimensions 10, 50 and 100. We will use the same setup as the previous section, except that the true subspace has only three nonzero coordinates. Because of the sample size that is required to estimate an active subspace in high dimension, it is computationally infeasible to deploy the GP or PRA active subspace estimation methods. Figure 6 shows the results of this analysis. The TBAS estimator is able to achieve much better accuracy with a significantly smaller cost.

5.5 Dimension Reduction with the Active Subspace

We now turn to the qualitative benefits of performing an active subspace analysis using TBAS. We used the NHEFS dataset of biochemistry tape and mortality data which were studied by Lundberg et al. (2019). This consisted of a dataset of 14,407 observations and 90 complete variables. We fit a TBAS to this dataset using a regression tree of depth 15, requiring at least 10 samples per leaf. We subsequently performed an eigendecomposition of the estimated active subspace matrix. Like Lundberg et al. (2019), we found that age was the most important variable and it mapped cleanly onto the first active subspace dimension. However, we found that the next most important dimension consisted of many variables (see Appendix B.3), such that they would have been difficult to capture with a typical variable-importance analysis. Figure 7 shows the result of this active subspace analysis, which reveals that after the second eigenvalue there appears to be a gap in the spectrum of the active subspace matrix, indicating the presence of an active subspace of dimension two (left panel). The middle panel shows a projection of the data, which breaks into two clusters along the second principal component, while the right panel shows a projection of randomly sampled points to visualize the predictive surface of the function. The right panel reveals the predictive surface has a fairly simple, "S-shaped" form over these two dimension. It would not be possible to detect this by considering only main effects or two factor interactions.

6 Discussion

Summary: We studied a simple method for estimating gradients in sufficiently deep regression trees on large datasets. Subsequently, we demonstrate how these estimates could be used to calculate active subspaces and integrated gradients. We found significant improvement in predictive performance could be achieved by using the tree-based active subspace to rotate the space, and also found that these estimates executed quicker than existing gradient-free active subspace estimators and had useful inductive bias towards detecting coordinate-sparse subspaces. Finally, we used these gradient estimates to reveal the simple global structure of a tree-based model fit to a complex dataset as well as the mechanism by which a random forest conducted MNIST classification.

Limitations: When it comes to limitations of our experiments, when comparing our active subspace estimates to the DASM, we used a particular neural network architecture and optimizer. It is possible that a different configuration would have performed better. Additionally, when it comes to limitations of the method we studied, our analysis suggests that it would take a very deep regression tree to estimate gradients to high accuracy in high dimension.

Conclusions: We think this work offers two high level conclusions. First, that trees may have a place in the field of Uncertainty Quantification, which has more commonly used differentiable but slow to estimate surrogates such as Gaussian processes and neural networks. Secondly, we hope that this work can also enable improve cross-pollination between developments in interpretability for neural networks and for regression trees by creating analogs for gradient-based neural network techniques.

We also would like to comment on how TBAS fit into the existing literature on interpretability in trees. It is somewhat distinct from the axiomatic approach to interpretability proffered by SHAP (Lundberg and Lee, 2017) and Integrated Gradients (Sundararajan et al., 2017). While an axiomatic approach can be useful, its universality should not be exaggerated. Hancox-Li and Kumar (2021) write about how it is unlikely that any variable selection method could satisfy all possible demands, no matter how attractive its axiomatic foundation. Since the active subspace is parameterized by a measure, it actually consists of an entire family of analyses, with different choices of emphasis over the input space possibly leading to different conclusions.

Future Work: We are excited about the future work opened up by the here-proposed gradient estimates. First, we suspect that part of the reason for the slow convergence rate of the gradient estimator is that we "start from scratch" at every depth of the tree, overwriting the previous estimator. Perhaps an improved convergence rate could be obtained by combining the gradient estimate from multiple depths rather than overwriting it. Next, while we have conducted some preliminary analyses in Appendix C.5, more work is needed to understand the properties of these estimates in classification problems with categorical inputs and missing data. We are also interested in the possibility of estimating higher order derivatives from tree structure. Furthermore, extending the class of functions approximated to nondifferentiable functions is of interest: does the quantity proposed here converge in such a case, and to what? But we are most excited about the potential to bring in other gradient-based techniques to tree-based methods. For instance, physics-informed machine learning (e.g. Raissi et al. (2019, 2017)) has been making waves in the fluid dynamics community (Cai et al., 2021) among other applications, where they allow the analyst to use information about function derivatives to improve predictions of differentiable models such as neural networks or Gaussian processes. Our gradient estimator opens up the possibility of deploying this technique in the context of regression trees.

Acknowledgments and Disclosure of Funding

I gratefully acknowledge funding from the NSF and NGA via DMS#2428033 as well as the NSF and AFOSR via DMS#2529277. The computational resources for this work were provided by the Unity Research Computing Platform, a multi-institutional cluster lead by University of Massachusetts Amherst, the University of Rhode Island, and University of Massachusetts Dartmouth. This project was inspired by discussions at the 2024 AISTATS conference.

References

Antoniadis, A., Lambert-Lacroix, S., and Poggi, J.-M. (2021). Random forests for global sensitivity analysis: A selective review. *Reliability Engineering & System Safety*, 206:107312.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Breiman, L. (2002). Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*, 1(58):3–42.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E. (2021). Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738.

Cai, Y., Ma, Y., Dong, Y., and Yang, H. (2023). Extrapolated random tree for regression. In *International Conference on Machine Learning*, pages 3442–3468. PMLR.

Chaudhuri, P., Lo, W.-D., Loh, W.-Y., and Yang, C.-C. (1995). Generalized regression trees. *Statistica Sinica*, pages 641–666.

Chen, H., Janizek, J. D., Lundberg, S., and Lee, S.-I. (2020). True to the model or true to the data? *arXiv preprint arXiv:2006.16234*.

Constantine, P. G. (2015). *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM.

Edeling, W. (2023). On the deep active-subspace method. *SIAM/ASA Journal on Uncertainty Quantification*, 11(1):62–90.

Elie-Dit-Cosaque, K. and Maume-Deschamps, V. (2024). Random forest based quantile-oriented sensitivity analysis indices estimation. *Computational Statistics*, 39(4):1747–1777.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Gao, W., Xu, F., and Zhou, Z.-H. (2022). Towards convergence rate analysis of random forests for classification. *Artificial Intelligence*, 313:103788.

Hancox-Li, L. and Kumar, I. E. (2021). Epistemic values in feature importance methods: Lessons from feminist epistemology. In *proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 817–826.

Hechtlinger, Y. (2016). Interpretation of prediction models using the input gradient. *arXiv preprint arXiv:1611.07634*.

Hokanson, J. M. and Constantine, P. G. (2018). Data-driven polynomial ridge approximation using variable projection. *SIAM Journal on Scientific Computing*, 40(3):A1566–A1589.

Hooker, G., Mentch, L., and Zhou, S. (2021). Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31:1–16.

Ishwaran, H. (2007). Variable importance in binary regression trees and forests.

Karczmarz, A., Michalak, T., Mukherjee, A., Sankowski, P., and Wygocki, P. (2022). Improved feature importance computation for tree models based on the banzhaf value. In *Uncertainty in Artificial Intelligence*, pages 969–979. PMLR.

Kazemitabar, J., Amini, A., Bloniarz, A., and Talwalkar, A. S. (2017). Variable importance using decision trees. *Advances in neural information processing systems*, 30.

Klusowski, J. (2021). Sharp analysis of a simple model for random forests. In *International Conference on Artificial Intelligence and Statistics*, pages 757–765. PMLR.

Klusowski, J. M. and Tian, P. M. (2020). Nonparametric variable screening with optimal decision stumps. *arXiv preprint arXiv:2011.02683*.

Li, X., Wang, Y., Basu, S., Kumbier, K., and Yu, B. (2019). A debiased mdi feature importance measure for random forests. *Advances in Neural Information Processing Systems*, 32.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23.

Louppe, G. (2014). Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2019). Explainable ai for trees: From local explanations to global understanding. *arXiv preprint arXiv:1905.04610*.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Ma, Y., Zhang, H., Cai, Y., and Yang, H. (2023). Decision tree for locally private estimation with public data. *Advances in Neural Information Processing Systems*, 36:43676–43705.

McCloskey, K., Taly, A., Monti, F., Brenner, M. P., and Colwell, L. J. (2019). Using attribution to decode binding mechanism in neural network models for chemistry. *Proceedings of the National Academy of Sciences*, 116(24):11624–11629.

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

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). Machine learning of linear differential equations using gaussian processes. *Journal of Computational Physics*, 348:683–693.

Raissi, M., Wang, Z., Triantafyllou, M. S., and Karniadakis, G. E. (2019). Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics*, 861:119–137.

Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630.

Sandri, M. and Zuccolotto, P. (2008). A bias correction algorithm for the gini variable importance measure in classification trees. *Journal of Computational and Graphical Statistics*, 17(3):611–628.

Sayres, R., Taly, A., Rahimy, E., Blumer, K., Coz, D., Hammel, N., Krause, J., Narayanaswamy, A., Rastegar, Z., Wu, D., et al. (2019). Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy. *Ophthalmology*, 126(4):552–564.

Shan, H., Zhang, J., and Kruger, U. (2015). Learning linear representation of space partitioning trees based on unsupervised kernel dimension reduction. *IEEE Transactions on Cybernetics*, 46(12):3427–3438.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC bioinformatics*, 9:1–11.

Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8:1–21.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Tripathy, R. and Bilionis, I. (2019). Deep active subspaces: A scalable method for high-dimensional uncertainty propagation. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 59179, page V001T02A074. American Society of Mechanical Engineers.

Vannucci, G., Siciliano, R., and Saltelli, A. (2024). Enhancing variable importance in random forests: A novel application of global sensitivity analysis. *arXiv preprint arXiv:2407.14194*.

Wen, H. and Hang, H. (2022). Random forest density estimation. In *International Conference on Machine Learning*, pages 23701–23722. PMLR.

Wycoff, N., Binois, M., and Gramacy, R. B. (2022). Sensitivity prewarping for local surrogate modeling. *Technometrics*, 64(4):535–547.

Wycoff, N., Binois, M., and Wild, S. M. (2021). Sequential learning of active subspaces. *Journal of Computational and Graphical Statistics*, 30(4):1224–1237.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286.

A Proofs

We begin by stating and discussing our assumptions.

A major difficulty in establishing theoretical results for tree-based methods is establishing the behavior of the thresholds as a function of the data generating process. For our purposes, the critical assumption is that the volume of the leaves shrink at an exponential rate in the number of splits made. We will study a scheme which achieves this by cycling between variables to split along at each depth, and subsequently splitting at the median of datapoints in each node along the split variable, such that about half of the data in the current node lie in each child node. To illustrate, the first split in the root node is made on variable 1, then at depth two, both child nodes have about half the data and themselves split along variable 2, and subsequently the four child nodes, each with a fourth of the data, split along variable 3. When we reach a depth of $P+1$, the 2^{P+1} splits will again be along variable 1, and each node has about $\frac{N}{2^{P+1}}$ many points in it.

This is of course typically not be how a regression tree is fit in the supervised setting, which will typically calculate splits taking y values into account. Establishing the convergence rate of the TBGE in such settings is left as future work. We believe that even the present proof, provided in the simplified setting, nevertheless provides valuable intuition as to the advantages and limitations of the TBGE.

Our first assumption concerns the splitting rate.

Assumption 0. *Let $S(N)$ denote the depth as a function of sample size.*

1. $\lim_{N\to\infty} S(N) = \infty$.
2. $\lim_{N\to\infty} \frac{S(N)}{\log N} = 0$.

This very slow rate of splitting is necessary to ensure that the number of datapoints in each node can grow to infinity while still eventually achieving arbitrarily small nodes.

Our next assumption is on the process which generates the feature variables.

Assumption 1. *The input points $\mathbf{x}_n$ are sampled from a distribution such that for any $\epsilon > 0$, there are constants $C_-, C_+ > 0$ and some N_ϵ such that for all $N > N_\epsilon$, we have:*

$$P\left(C_- 2^{-\frac{D_i}{P}} \leq |u_p^i - l_p^i| \leq C_+ 2^{-\frac{D_i}{P}}\right) > 1-\epsilon \quad \forall p \in \{1,\dots,P\},$$

where $D_i := 1 + \lfloor \log_2(i+1) \rfloor$ is the depth of node i.

This assumption tells us that the width of the leaf nodes will shrink exponentially. This assumption is most likely to be violated in observational contexts, where the feature variables may live on some manifold within $[0,1]^P$ and thus leave gaps within the unit cube unfilled. Naturally, the gradient estimates for regions without data will not behave well. However, this assumption may be satisfied in contexts where the feature variables may be set by the experimenter. In particular, this property holds for the uniform distribution, as we now discuss.

Remark 1. *Assumption 1 is satisfied if $\mathbf{x}_1, \dots, \mathbf{x}_N \overset{i.i.d}{\sim} U([0,1]^P)$ and $S(N) = \lfloor \log_2 \log_2 N \rfloor$.*

Proof. Start with the case $P = 1$, where we have uniform data on $[0,1]$, and are making $S(N)$ many recursive splits so as to keep an equal number of points in each bin. The distribution of the bin borders is given by the order statistic distributions. For uniform data, it is known that the difference between the j^{th} and k^{th} order statistics is given by a Beta distribution with $\alpha = k-j$ and $\beta = N+1-(k-j)$ when $k > j$. As each bin has $\frac{N}{2^{S(N)}}$ observations, $k - j = \frac{N}{2^{S(N)}}$ and the marginal distribution of a bin width is thus:

$$u^i - l^i \sim \text{Beta}\left(\frac{N}{2^{S(N)}}, N+1-\frac{N}{2^{S(N)}}\right).$$

We thus have that:

1. $\mathbb{E}[u^i - l^i] = \left(\frac{N}{N+1}\right)\frac{1}{2^{S(N)}}$,

2. $\mathbb{V}[u^i - l^i] = \left(\frac{N^2}{(N+1)^2(N+2)}\right)\left(1 + \frac{1}{N} - \frac{1}{2^{S(N)}}\right)\frac{1}{2^{S(N)}} := M\frac{1}{N2^{S(N)}}$,

where $M = \frac{N^3}{(N+1)^2(N+2)}\left(1 + \frac{1}{N} - \frac{1}{2^{S(N)}}\right)$, which is bounded above by 1 for all $N \geq 1$.

We next develop a Chebyshev bound, valid for all $k > 1$:

$$P\left(\left|(u^i - l^i) - \frac{N}{N+1}\frac{1}{2^{S(N)}}\right| \leq \frac{kM}{\sqrt{N2^{S(N)}}}\right) \geq 1 - \frac{1}{k^2}$$
$$\iff P\left(\left|2^{S(N)}(u^i - l^i) - \frac{N}{N+1}\right| \leq kM\sqrt{\frac{2^{S(N)}}{N}}\right) \geq 1 - \frac{1}{k^2}\,.$$

We now fix some $\epsilon > 0$, and set $k = \frac{1}{\sqrt{\epsilon}}$, such that:

$$P\left(\left|2^{S(N)}(u^i - l^i) - \frac{N}{N+1}\right| \leq M\sqrt{\frac{2^{S(N)}}{N}}\frac{1}{\sqrt{\epsilon}}\right) \geq 1 - \epsilon$$
$$\implies P\left(\left|2^{S(N)}(u^i - l^i) - \frac{N}{N+1}\right| \leq \sqrt{\frac{2^{S(N)}}{N}}\frac{1}{\sqrt{\epsilon}}\right) \geq 1 - \epsilon\,,$$

where in moving between the lines we have dropped the M term as $M \leq 1$.

This gives us that, with high probability,

$$\left|2^{S(N)}(u - l) - \frac{N}{N+1}\right| \leq \sqrt{\frac{2^{S(N)}}{N\epsilon}}$$
$$\iff -\sqrt{\frac{2^{S(N)}}{N\epsilon}} \leq 2^{S(N)}(u - l) - \frac{N}{N+1} \leq \sqrt{\frac{2^{S(N)}}{N\epsilon}}$$
$$\iff -\sqrt{\frac{2^{S(N)}}{N\epsilon}} + \frac{N}{N+1} \leq 2^{S(N)}(u - l) \leq \sqrt{\frac{2^{S(N)}}{N\epsilon}} + \frac{N}{N+1}$$
$$\iff 2^{-S(N)}\left(\frac{N}{N+1} - \sqrt{\frac{2^{S(N)}}{N\epsilon}}\right) \leq (u - l) \leq 2^{-S(N)}\left(\frac{N}{N+1} + \sqrt{\frac{2^{S(N)}}{N\epsilon}}\right)\,.$$

The quantity $\frac{2^{S(N)}}{N} = \frac{\log_2(N)}{N}$, viewed as a continuous function of N, achieves its maximum value of $\frac{1}{e\log(2)}$ at $N = e$, and $\frac{N}{N+1}$ is bounded above by 1 therefore:

$$2^{-S(N)}\left(\frac{N}{N+1} + \sqrt{\frac{2^{S(N)}}{N\epsilon}}\right) \leq 2^{-S(N)}\left(1 + \sqrt{\frac{1}{e\log(2)\epsilon}}\right)\,.$$

Using the fact that $\frac{N}{N+1}$ is bounded below by $\frac{1}{2}$ for $N \geq 1$ and plugging in the bound expression for $S(N)$ yields:

$$2^{-S(N)}\left(\frac{N}{N+1} - \sqrt{\frac{2^{S(N)}}{N\epsilon}}\right) \geq 2^{-S(N)}\left(\frac{1}{2} - \sqrt{\frac{\log_2(N)}{N\epsilon}}\right)\,.$$

Choosing N large enough that $\frac{\log N}{N} < \frac{\epsilon^2}{4}$ will ensure that $\frac{1}{2} - \sqrt{\frac{\log_2(N)}{N\epsilon}} := L$ is positive.

We thus arrive at the following with high probability:

$$2^{-S(N)}L \leq u - l \leq 2^{-S(N)}\left(1 + \sqrt{\frac{1}{e\log(2)\epsilon}}\right)$$

When $P > 1$, we alternate between splitting along each variable. The marginal distribution of x_p remains uniform within each split node, which allows us to reduce the general dimensional case to the one dimensional case. However, each variable is split only every P^{th} pass, which rescales the $S(N)$ term in the exponent by $\frac{1}{P}$. □

Our final assumptions are on the function and error process.

Assumption 2. *Let $y_n = f(\mathbf{x}_n) + \epsilon_n$. We assume that:*

1. *The estimate of the mean of $f(\mathbf{x})$ over some interval $[\mathbf{a}, \mathbf{b}]$ converges at the usual square root rate; this is achieved for example if the error terms $\epsilon_1, \ldots, \epsilon_N$ are such that $\mathbb{E}[\epsilon_n] = 0$; $\mathbb{V}[\epsilon_n] \leq \infty$ and $\epsilon_{n_1} \perp\!\!\!\perp \epsilon_{n_2} \forall n_1, n_2 \in \{1, \ldots, N\}$.*
2. *f is twice differentiable with $\|\nabla^2 f(\mathbf{x})\| \leq H \ \forall \mathbf{x} \in [0,1]^P$, i.e. f has a Hessian with operator norm bounded by H.*

The first assumption gives us that the sample means in a given node converge to the integral of the function over that node normalized by the node's volume. The second assumption allows us to bound the error of a 1st order Taylor approximation.

We are now prepared to prove our results.

Theorem 4.1. *Let $S(N)$ denote the number of splits in the tree as a function of the sample size. Let $\frac{\tilde{\partial}^N f(\mathbf{x})}{\tilde{\partial} x_p}$ denote the TBGE at point $\mathbf{x} \in [0,1]^P$ with sample size N. Under Assumptions 0, 1, and 2:*

$$\frac{\tilde{\partial}^N f(\mathbf{x})}{\tilde{\partial} x_p} = \frac{\partial f(\mathbf{x})}{\partial x_p} + O_P\left(\frac{2^{\frac{P+2}{2P}S(N)}}{\sqrt{N}} + P2^{-\frac{S(N)}{P}}\right).$$

Proof. At each node i, the quantity γ_i serves as our estimate of the partial derivative in the σ_i direction near that node. For a fixed i, we do not expect convergence of γ_i to any gradient quantity. However, we do expect convergence as we go down the tree depth for suitably chosen $S(N)$.

We wish to estimate the p partial derivative at some location $\mathbf{x} \in [0,1]^P$ given a sample of size N. According to our simplified splitting scheme which alternates between variables, and given the depth $S(N) > P$, the most recent split along variable p is given by the deepest node i of depth K no greater than $S(N)$ which splits along variable p (that is, $i \equiv p \pmod P$) and which contains $\mathbf{x}$ (that is, $B^K(\mathbf{x}) = i$). Our task is to show that this sequence of γ_i, with i thusly viewed implicitly as a function of N, will converge to $\frac{\partial f(\mathbf{x})}{\partial x_p}$.

Recall the definition of the finite-difference like estimate:

$$\gamma_i = \frac{2(\hat{\mu}_r^i - \hat{\mu}_l^i)}{u_{\sigma_i}^i - l_{\sigma_i}^i}, \tag{9}$$

and $\sigma_i = p$ by our choice of i.

$\hat{\mu}_l^i$ is the mean of the data falling within the left child of node i. Assumption 2.1 gives us that within any rectangle, the sample mean converges to the population mean at the usual square root rate as a function of the number of points in that rectangle. In the case of the our simplified splitting scheme, the number of points in a cell is given by the total number of points divided by the number of cells, or by $\frac{N}{2^{S(N)}}$. Hence we have that:

$$\hat{\mu}_l^i = \mathop{\mathbb{E}}_{\mathbf{x} \sim U\left([\mathbf{l}^{c_l^i}, \mathbf{u}^{c_l^i}]\right)}[f(\mathbf{x})] + O_P\left(\sqrt{\frac{2^{S(N)}}{N}}\right) := \mu_l^i + O_P\left(\sqrt{\frac{2^{S(N)}}{N}}\right), \tag{10}$$

where the second equation serves to give our definition of μ_l^i, a quantity independent of our dataset and instead dependent on f, the mean function of the relationship being approximated. Here, the expectation is taken with respect to the uniform distribution over the bounds of the left child node of i, which as before we denote by c_l^i. We of course have a similar definition for the right child node's mean μ_r^i.

Next we approximate f at an arbitrary point $\mathbf{z} \in [\mathbf{l}^i, \mathbf{u}^i]$ by expanding it about $\mathbf{x}$ (or any other point in $[\mathbf{l}^i, \mathbf{u}^i]$), using the Lagrange form of the remainder with ξ dependent on $\mathbf{z}$:

$$f(\mathbf{z}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{z} - \mathbf{x}) + \frac{1}{2}(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi)(\mathbf{z} - \mathbf{x}), \tag{11}$$

and plug this into the definition of μ_l^i:

$$\mu_l^i = \mathop{\mathbb{E}}_{\mathbf{z}\sim U\left([\mathbf{l}^{c_l^i}, \mathbf{u}^{c_l^i}]\right)} [f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{z} - \mathbf{x}) + \frac{1}{2}(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi)(\mathbf{z} - \mathbf{x})]$$
$$= f(\mathbf{x}) + \frac{1}{2}\nabla f(\mathbf{x})^\top \left(\mathbf{l}^{c_l^i} + \mathbf{u}^{c_l^i}\right) - \nabla f(\mathbf{x})^\top \mathbf{x} + \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{z}\sim U\left([\mathbf{l}^{c_l^i}, \mathbf{u}^{c_l^i}]\right)} [(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi)(\mathbf{z} - \mathbf{x})] .$$

We needed only the linearity of expectation to evaluate the first two terms. Since we likewise have:

$$\mu_r^i = f(\mathbf{x}) + \frac{1}{2}\nabla f(\mathbf{x})^\top \left(\mathbf{l}^{c_r^i} + \mathbf{u}^{c_r^i}\right) - \nabla f(\mathbf{x})^\top \mathbf{u}^i + \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{z}\sim U\left([\mathbf{l}^{c_r^i}, \mathbf{u}^{c_r^i}]\right)} [(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi)(\mathbf{z} - \mathbf{x})] ,$$

taking the difference yields:

$$\mu_r^i - \mu_l^i = \nabla f(\mathbf{x})^\top \left(\mathbf{l}^{c_r^i} - \mathbf{l}^{c_l^i} + \mathbf{u}^{c_r^i} - \mathbf{u}^{c_l^i}\right) + R ,$$

where

$$R = \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{z}\sim U\left([\mathbf{l}^{c_r^i}, \mathbf{u}^{c_r^i}]\right)} [(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi(\mathbf{x}))(\mathbf{z} - \mathbf{x})] - \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{z}\sim U\left([\mathbf{l}^{c_l^i}, \mathbf{u}^{c_l^i}]\right)} [(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi(\mathbf{x}))(\mathbf{z} - \mathbf{x})] .$$

As siblings, the bounds of c_r^i and c_l^i agree except for along the p dimension, such that:

$$\nabla f(\mathbf{x})^\top \left(\mathbf{l}^{c_r^i} - \mathbf{l}^{c_l^i} + \mathbf{u}^{c_r^i} - \mathbf{u}^{c_l^i}\right)$$
$$= \sum_{k\neq p} \frac{\partial f(\mathbf{x})}{\partial x_k}(l_k^i - l_k^i + u_k^i - u_k^i) + \frac{\partial f(\mathbf{x})}{\partial x_p}(\tau^i - l_p^i + u_p^i - \tau^i) = \frac{\partial f(\mathbf{x})}{\partial x_p}(u_p^i - l_p^i) .$$

Therefore:

$$\mu_r^i - \mu_l^i = \frac{1}{2}\frac{\partial f(\mathbf{x})}{\partial x_p}(u_p^i - l_p^i) + R ,$$

such that

$$\frac{2(\mu_r^i - \mu_l^i)}{u_p^i - l_p^i} = \frac{\partial f(\mathbf{x})}{\partial x_p} + \frac{2R}{u_p^i - l_p^i} .$$

Let's now work to develop R. Note that:

$$\left| \mathop{\mathbb{E}}_{\mathbf{z}\sim U\left([\mathbf{l}^{c_r^i}, \mathbf{u}^{c_r^i}]\right)} [(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi(\mathbf{z}))(\mathbf{z} - \mathbf{x})] \right| \leq \max_{\mathbf{z}\in[\mathbf{l}^{c_r^i}, \mathbf{u}^{c_r^i}]} \left|(\mathbf{z} - \mathbf{x})^\top \nabla^2 f(\xi(\mathbf{z}))(\mathbf{z} - \mathbf{x})\right| \quad (12)$$

$$\overset{C.S.}{\leq} \max_{\mathbf{z}\in[\mathbf{l}^{c_r^i}, \mathbf{u}^{c_r^i}]} \|\mathbf{z} - \mathbf{x}\| \left\|\nabla^2 f(\xi(\mathbf{z}))(\mathbf{z} - \mathbf{x})\right\| \overset{\text{def } \|\cdot\|}{\leq} \max_{\mathbf{z}\in[\mathbf{l}^{c_r^i}, \mathbf{u}^{c_r^i}]} \|\mathbf{z} - \mathbf{x}\|^2 \left\|\nabla^2 f(\xi(\mathbf{z}))\right\| . \quad (13)$$

The first inequality comes from the absolute integral being less than the max of the absolute integrand, the second comes from Cauchy-Schwartz, and the third from the definition of the operator norm. Bounding the max of the product of positive quantities by the product of the maxes and invoking our bound on $\left\|\nabla^2 f(\xi(\mathbf{z}))\right\|$ will be our next steps, yielding:

$$\max_{\mathbf{z}\in[\mathbf{l}^{c_i^r}, \mathbf{u}^{c_i^r}]} \|\mathbf{z} - \mathbf{x}\|^2 \left\|\nabla^2 f(\xi(\mathbf{z}))\right\| \leq H \left\|\mathbf{u}^i - \mathbf{l}^i\right\|^2 .$$

Making similar moves for the left child node gives us that:

$$\frac{2|R|}{u_p^i - l_p^i} \leq H \frac{2\|\mathbf{u}^i - \mathbf{l}^i\|^2}{u_p^i - l_p^i} .$$

Under our assumptions on the splitting process, we have that $C_- 2^{-\frac{S(N)}{P}} \leq |u_k^i - l_k^i| \leq C_+ 2^{-\frac{S(N)}{P}}$ for some $C_-, C_+ > 0$ with high probability. Thus, also with high probability:

$$\frac{\|\mathbf{u}^i - \mathbf{l}^i\|^2}{u_p^i - l_p^i} \leq \frac{PC_+^2 2^{-2\frac{S(N)}{P}}}{C_- 2^{-\frac{S(N)}{P}}} = \left(P\frac{C_+^2}{C_-}\right) 2^{-\frac{S(N)}{P}},$$

and therefore

$$\frac{2(\mu_r^i - \mu_l^i)}{u_p^i - l_p^i} = \frac{\partial f(\mathbf{x})}{\partial x_p} + O_P\left(P 2^{-\frac{S(N)}{P}}\right).$$

We thus have a bound on the difference between the true functional mean and the gradient, but have thus far been ignoring error in the estimation of these means. To bring this into the equation, we begin by noting that, with high probability:

$$\hat{\mu}_r^i = \mu_r^i + O_P\left(\sqrt{\frac{2^{S(N)}}{N}}\right) \implies \frac{\hat{\mu}_r^i}{u_p^i - l_p^i} = \frac{\mu_r^i}{u_p^i - l_p^i} + O_P\left(\sqrt{\frac{2^{\frac{P+2}{P}S(N)}}{N}}\right).$$

This is because for large enough N, $u_p^i - l_p^i \geq C_- 2^{\frac{-S(N)}{P}}$ with high probability. and similarly for $\hat{\mu}_l^i$, which gives us that:

$$\frac{2(\hat{\mu}_r^i - \hat{\mu}_l^i)}{u_p^i - l_p^i} = \frac{2(\mu_r^i - \mu_l^i)}{u_p^i - l_p^i} + O_P\left(\frac{2^{\frac{P+2}{2P}S(N)}}{\sqrt{N}}\right),$$

Combining with our earlier work yields:

$$\frac{2(\hat{\mu}_r^i - \hat{\mu}_l^i)}{u_p^i - l_p^i} = \frac{\partial f(\mathbf{x})}{\partial x_p} + O_P\left(\frac{2^{\frac{P+2}{2P}S(N)}}{\sqrt{N}} + P2^{-\frac{S(N)}{P}}\right).$$

□

Now let $S(N) = P \log\log N$. Then:

$$\frac{2^{\frac{P+2}{2P}S(N)}}{\sqrt{N}} + P2^{-\frac{S(N)}{P}} = \frac{\left(2^{P\log\log N}\right)^{\frac{1}{2}+\frac{1}{P}}}{\sqrt{N}} + P2^{-\log\log N} = \frac{(\log N)^{\frac{P}{2}+1}}{\sqrt{N}} + \frac{P}{\log N}.$$

Since the first term vanishes relative to the second, we obtain:

$$\frac{2(\hat{\mu}_r^i - \hat{\mu}_l^i)}{u_p^i - l_p^i} = \frac{\partial f(\mathbf{x})}{\partial x_p} + O_P\left(\frac{P}{\log N}\right).$$

Theorem 4.2. *Let $h : \mathbb{R}^P \to \mathbb{R}^P$ be bounded in the sense that $h(\mathbf{z}) \leq C_h \|\mathbf{z}\|^{a_h}$ for $a_h > 0$. Under Assumptions 0, 1 and 2, the Partition-Based estimator converges to the true integro-differential quantity as the observation sample size diverges. That is,*

$$\lim_{N\to\infty} \hat{\mathcal{I}}_{PB}(f) = \lim_{N\to\infty} \sum_{i\in\mathcal{N}_{K-1}} h(\mathbf{G}^i)\mu([\mathbf{l}^i, \mathbf{u}^i]). = \int h(\nabla f(\mathbf{x}))d\mu(\mathbf{x}) = \mathcal{I}(f).$$

Proof. Since the function sequence $\mathbf{g}^k(\mathbf{x}) := G_{B^k(\mathbf{x})}$ converges pointwise to $\nabla f(\mathbf{x})$ by Proposition 1, we need only establish a function $H(\mathbf{x})$ which dominates $\mathbf{g}^k(\mathbf{x})$ and apply the Dominated Convergence Theorem.

To this end, denote by C^r, C^l the child nodes of node i and examine its gradient estimator's pth entry, given by:

$$\frac{2(\mu_i^r - \mu_i^l)}{u_p^i - l_p^i} \overset{N\to\infty}{\to} \frac{2\left(\frac{1}{|\mathcal{N}_{C^r}|}\int_{\mathcal{N}_{C^r}} f(\mathbf{x})d\mathbf{x} - \frac{1}{|\mathcal{N}_{C_i^l}|}\int_{\mathcal{N}_{C_i^l}} f(\mathbf{x})d\mathbf{x}\right)}{u_p^i - l_p^i}. \tag{14}$$

The magnitude of this difference in averages is bounded by the magnitude of the difference of extremes:

$$|\frac{1}{|\mathcal{N}_{C_i^r}|}\int_{\mathcal{N}_{C_i^r}} f(\mathbf{x})d\mathbf{x} - \frac{1}{|\mathcal{N}_{C_i^r}|}\int_{\mathcal{N}_{C_i^l}} f(\mathbf{x})d\mathbf{x}| \leq \max_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathcal{N}_{C_i^r}\times\mathcal{N}_{C_i^l}} |f(\mathbf{x}_1) - f(\mathbf{x}_2)|\,. \tag{15}$$

But since f is continuously differentiable, it is also Lipschitz continuous (call the constant L), and we have that:

$$\max_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathcal{N}_{C_i^r}\times\mathcal{N}_{C_i^l}} |f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq PL\|\mathbf{x}_1 - \mathbf{x}_2\|_\infty\,. \tag{16}$$

Therefore:

$$\left|\frac{2(v_{C_i^r} - v_{C_i^l})}{u_p^i - l_p^i}\right| \leq \left|\frac{2(PL\|\mathbf{x}_1 - \mathbf{x}_2\|_\infty)}{u_p^i - l_p^i}\right| \leq 2PL\,. \tag{17}$$

Hence $\|g^k(\mathbf{x})\|_2 \leq 2P^2L$, , and thence $C_h(2P^2L)^{a_h}$ bounds $h(g^k(\mathbf{x}))$. Since the integral is over the unit hypercube, the constant function is integrable and we can apply the Dominated Convergence Theorem to yield the desired result.

□

Theorem A.1. *Under Assumptions 1 and 2, the Monte Carlo estimator converges to the true integro-differential quantity as the Monte Carlo sample size and the observation sample size diverge. That is,*

$$\lim_{N,M\to\infty} \hat{\mathcal{I}}_{MC}(f) = \lim_{N,M\to\infty} \frac{1}{M}\sum_{\mathbf{x}_m\sim\mu} h(\tilde{\nabla}f(\mathbf{x})) = \int h(\nabla f(\mathbf{x}))d\mu(\mathbf{x}) = \mathcal{I}(f)\,.$$

Proof. This follows from the fact that

$$\lim_{N\to\infty}\lim_{M\to\infty} \frac{1}{M}\sum_{\mathbf{x}_m\sim\mu} h(\tilde{\nabla}f(\mathbf{x}_m)) = \tag{18}$$

$$= \lim_{N\to\infty}\int_{\mathbf{x}\in[0,1]^P} h(\tilde{\nabla}f(\mathbf{x}))d\mu = \lim_{N\to\infty}\sum_{i\in\mathcal{D}_k} h(\mathbf{G}_i)\mu([\mathbf{l}^i,\mathbf{u}^i]) \tag{19}$$

and an application of Theorem 4.2. □

A.1 Implications for Active Subspaces and Integrated Gradients

We conclude this section by explicitly stating the implications of these results for Tree-based Active Subspace (TBAS) estimation.

Corollary A.2. *Let $K(N)$ denote the depth of the regression tree as a function of N. Under Assumptions 1 and 2, we have that*

$$\lim_{N\to\infty}\sum_{i\in\mathcal{D}_{K(N)-1}} \mathbf{G}_i\mathbf{G}_i^\top\mu(\mathcal{N}_i) = \int_{[0,1]^P}\nabla f(\mathbf{x})\nabla f(\mathbf{x})d\mu(\mathbf{x})\,.$$

Proof. Follows from Theorem 4.2. □

As well as for Tree-Based Integrated Gradient (TBIG) estimation.

Corollary A.3. *Let $K(N)$ denote the depth of the regression tree as a function of N. Under Assumptions 1 and 2, we have that, assuming that u_n are iid uniform on [0,1]:*

$$\lim_{k,N,M\to\infty}(\mathbf{x}-\mathbf{x}^*)\odot\frac{1}{M}\sum_{m=1}^{M}\tilde{\nabla}f\left(u_m\mathbf{x}+(1-u_m)\mathbf{x}^*\right)$$
$$= (\mathbf{x}-\mathbf{x}^*)\odot\int_{\alpha=0}^{1}\nabla f(\alpha\mathbf{x}+(1-\alpha)\mathbf{x}^*)d\mathbf{x}\,.$$

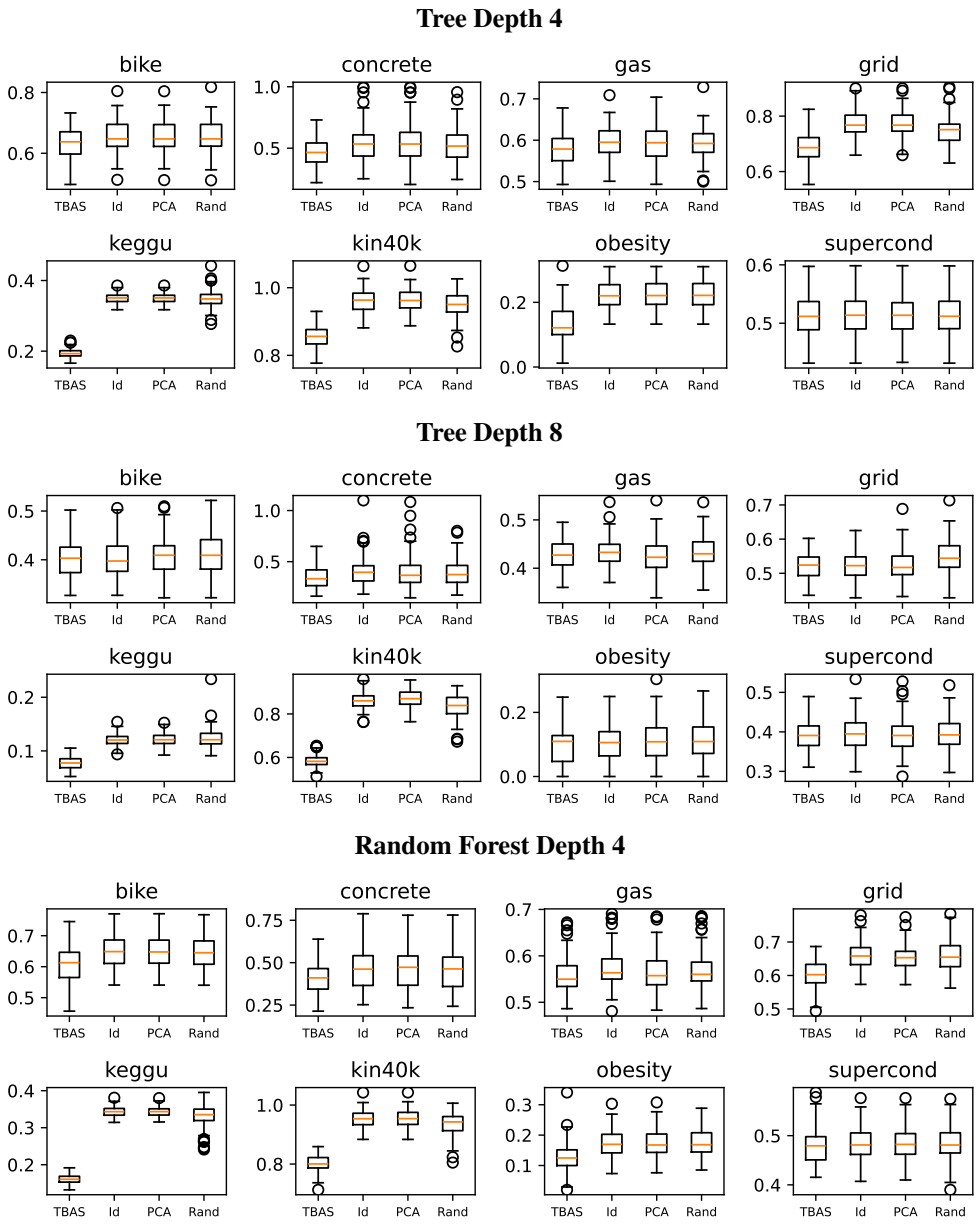

Figure 8: Boxplots corresponding to Table 1.

Proof. Follows from Theorem A.1. □

B Details of Numerical Experiments

This section gives additional details and discussion of the numerical results presented in Section 5. All tree-based models are estimated using Scikit-Learn Pedregosa et al. (2011).

B.1 Rotation Prediction Study Additional Details

In this study, when performing a rotation, we used only the $\sqrt{P}$ many PCA components or Active Subspace dimensions, and appended these to the original design matrix as new variables. We measure prediction error using 100-fold cross validation.

The table below gives the parameters of the datasets used for the prediction study of Section 5.2.

Name	N	P	URL
concrete	1,030	9	https://archive.ics.uci.edu/dataset/165/
kin40k	40,000	9	https://github.com/alshedivat/keras-gp/kgp/datasets/kin40k.py
keggu	65,554	28	https://www.genome.jp/kegg/pathway.html
bike	17,379	13	https://archive.ics.uci.edu/dataset/560/
obesity	2,111	24	https://archive.ics.uci.edu/dataset/544/
gas	36,733	12	https://archive.ics.uci.edu/dataset/224
grid	10,000	13	https://archive.ics.uci.edu/dataset/471/
supercond	21,263	82	https://archive.ics.uci.edu/dataset/464/

We also provide boxplots of Cross Validation errors in Figure 8. Running this study took about five hours on a 40 core Ubuntu machine with 128 GB of RAM.

B.2 Active Subspace Estimation Study Additional Details

In Section 5.3, We randomly sampled a unit vector $\mathbf{a}$ from the uniform distribution over directions and then sampled input points uniformly at random on the unit cube. We subsequently evaluated the function $f(\mathbf{x}) = \cos(6\pi(\mathbf{a}^\top(\mathbf{x} - 0.5))$ which was treated as the noiseless observed response $\mathbf{y}$. We compared the estimates with using the angle each made with $\mathbf{a}$. This experiment was repeated 20 times.

We used the implementation of PRA provided with the pypi package `PSDR`[1]. For GP-based active subspace estimation, we used the CRAN package `activegp`. We used all default settings for the PRA method as well as for the GP method. For the DASM, we used a neural network with an additional layer of width 512 subsequent to the active subspace layer, and used gradient descent with a step size of 10^{-3} on the Mean Squared Error cost function. This neural network was implemented in JAX Bradbury et al. (2018). In addition to the quantitative advantages enjoyed by the tree-based method of active subspace estimation, we would also like to note that like the GP-based method, and unlike the PRA and DASM, it provides an estimate of the entire active subspace matrix, rather than simply a basis for the active subspace. This is important for two reasons. First, with the active subspace matrix in hand, we can create analogs of PCA scree plots to determine what dimension of the active subspace is most desirable, or to get some idea of how much information is being lost in, say, a two dimensional visualization. And secondly, it allows us to decide on an active subspace dimension *after* having seen the data rather than before, without requiring the estimation procedure to be re-run.

Running this study took about eight hours on a 40 core Ubuntu machine with 128 GB of RAM.

B.3 NHEFS Data Analysis Details

This table presents the first 3 eigenvectors of the mortality data analysis, restricting to the top 9 variables with highest coefficients. The first eigenvector captures almost entirely the age variable, which Lundberg et al. (2019) also found to be most important. We see that the second eigenvector is evenly distributed across sex and one of the urine variables. The third eigenvector is dominated by the urineDark variable.

Eigenvector	age	urineDark	sex	urineNeg	SGOT	hemoglobin	urineAlb	total	physical
1	-1.00	-0.00	0.01	0.01	-0.00	-0.00	0.00	-0.00	-0.00
2	0.01	-0.15	0.56	0.53	-0.28	-0.30	0.30	-0.16	0.09
3	-0.00	-0.96	-0.03	-0.11	0.16	0.07	0.02	0.04	0.07

When producing the right panel of Figure 7, we used the prediction after accounting for the effect of age in order to demonstrate the change in the predictive surface over the second and third eigenvalues.

[1] https://psdr.readthedocs.io/en/latest/

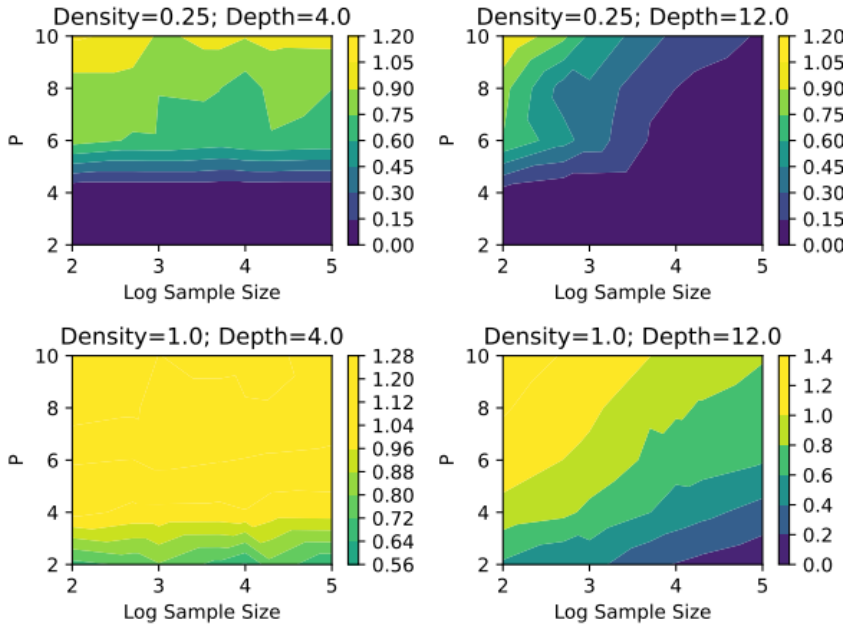

Figure 9: Convergence of Gradient Estimates in Sample Size.

C Additional Numerical Experiments

C.1 Empirical Gradient Estimate Quality

We present the results of a simulation showing the empirical performance of TBGE in estimating gradients under various tree depth, sample sizes, dimensions and gradient densities on the function $f(x) = \log(1 + \mathbf{a}^\top \mathbf{x})$ with nonzero elements of $\mathbf{a}$ generated from an iid Gaussian. Figure 9 contains heatmaps showing the results for all combinations of a 25% and 100% dense gradient and a depth 4 and 12 tree. In each figure, the x-axis gives the sample size and the y-axis the problem dimension. The color represents the angle between the true and estimated gradient. We see that a deep tree is able to decrease error as sample size increases while a shallow one cannot, and also that error decreases much faster for the sparse gradient.

C.2 Gradient Error Comparison against GP Surrogate.

In this section we conduct a study similar to Section 5.3, but focus directly on the gradient quality rather than the active subspace quality. That is, we compare the accuracy of gradient estimates produced by a regression tree against those provided by a Gaussian process for a fixed *compute time*, rather than a fixed dataset size. The idea is that on simpler simulations, we may be able to produce an abundance of data and that trees might be useful in this scenario. In Figure 10, we find that indeed the tree can produce useful estimates extremely quickly for larger sample sizes, leading to some comparative advantage compared to a GP fit on a smaller sample size (but taking even longer to make predictions).

C.3 Active Subspace Estimation Performance under Noise

This section repeats the experiments of Section 5.3 where the output data are contaminated under small i.i.d. Gaussian noise with standard deviation 0.1. The results, presented in Figure 11 are very similar to the noiseless case but are nevertheless included for completeness.

C.4 Gradient Estimate under Input Variable Correlation

Our numerical experiments on real data suggest our proposed method can operate under correlated inputs, and in this section we conduct an experiment to more closely investigate quantitatively the

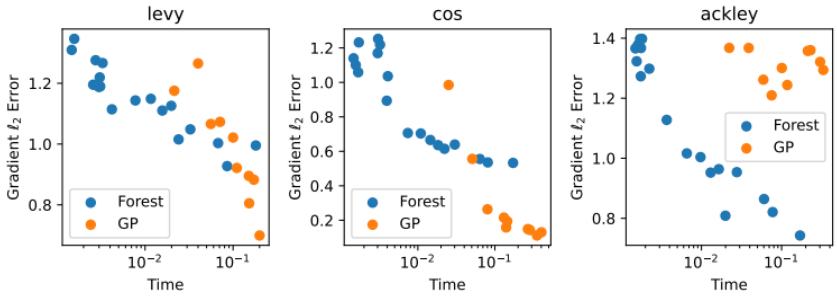

Figure 10: Gradient Error Comparison against GP surrogate.

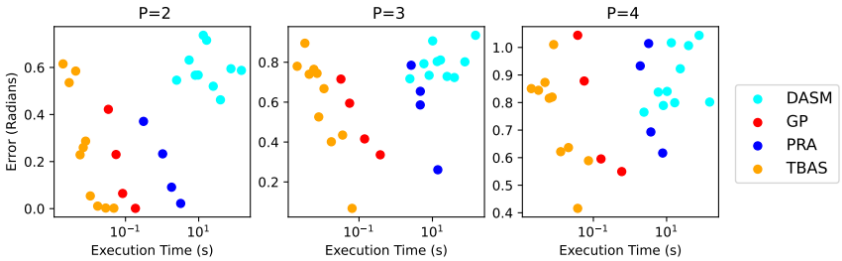

Figure 11: Active Subspace Estimation with Gaussian Noise (std=0.1;iid).

impact of input-variable correlation on gradient estimate quality. Sampling data from a truncated normal in 5D with correlation varying from 0 to 0.99, we evaluated estimates of the Ackley function's gradient at 100 random points. The results are presented in Figure 12. We see that correlation can have some deleterious effects on the gradient estimates; however, it requires a very high level of correlation before the estimates are significantly affected.

C.5 Numerical Experiments on Classification Trees

We repeat the experiments of Section 5.2, but now by replacing each regression problem with a classification problem by assessing whether a given observation falls above or below the median observation. The results are given in Figure 13. Intriguingly, the results are significantly less

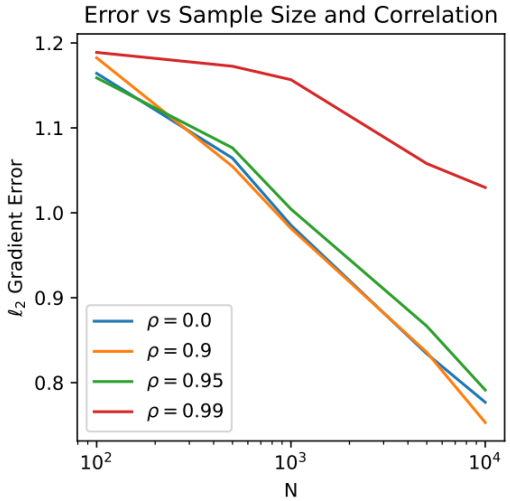

Figure 12: The Effect of Correlation on Gradient Estimates.

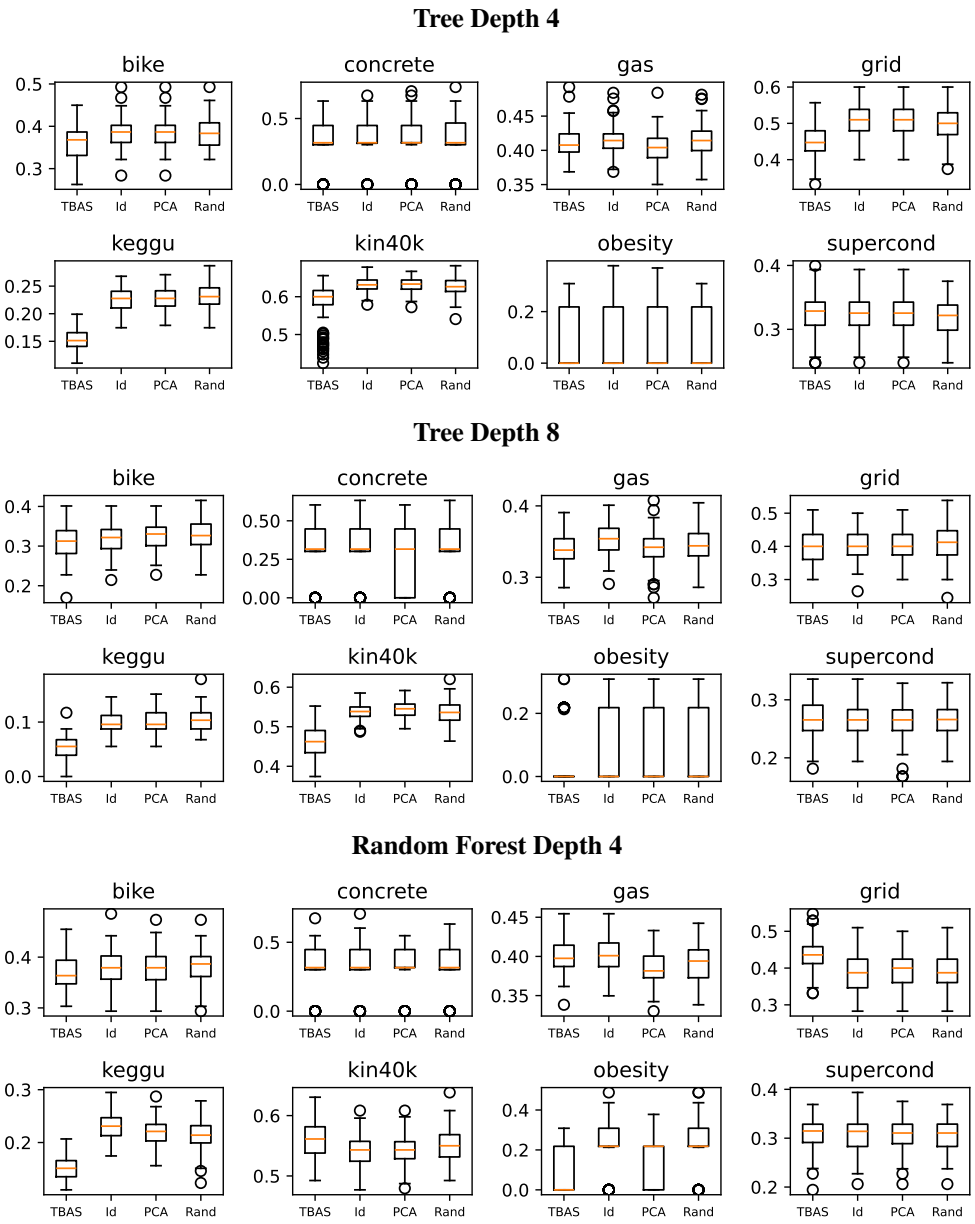

Figure 13: Classification Exercise: Brier Scores are indicated; lower is better.

promising for the TBAS method, despite the fact that by construction, there is structure in the data that TBAS could possibly exploit. This indicates that there may be special considerations to be taken in deploying this methodology to classification problems.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We propose a new way to get gradient information from regression trees, as we mention in the introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes we have a dedicated Limitations section in the discussion.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: We provide all proofs in the supplementary material, and carefully list our assumptions as part of Assumption 1 and Assumption 2 of the Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide all necessary details to reproduce our paper in the narrative and algorithms.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide code reproducing our results and a README with instructions for doing so.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide all necessary information when describing our runs.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: In our numerical results we use confidence interval overlap to assess significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide this in Appendix B.2.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes we respect this code of ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Yes we give discussion of this in the Discussion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This work has a low risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We give dataset information in Appendix B.1.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: This paper primarily introduces methodology.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No crowdsourcing or other human subjects whatsoever.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No crowdsourcing or other human subjects whatsoever.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: LLMs were not involved in any aspect of this project.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.