Enhance Learning Efficiency of Oblique Decision Tree via Feature Concatenation

Shen-Huan Lyu   LVSH@HHU.EDU.CN
Key Laboratory of Water Big Data Technology of Ministry of Water Resources,
College of Computer Science and Software Engineering, Hohai University, Nanjing, China
Department of Computer Science, City University of Hong Kong, Hong Kong, China
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Yi-Xiao He ✉   HEYX@NJUCM.EDU.CN
School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine,
Nanjing, China

Yanyan Wang   YANYAN.WANG@HHU.EDU.CN
Key Laboratory of Water Big Data Technology of Ministry of Water Resources,
College of Computer Science and Software Engineering, Hohai University, Nanjing, China
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Zhihao Qu   QUZHIHAO@HHU.EDU.CN
Key Laboratory of Water Big Data Technology of Ministry of Water Resources,
College of Computer Science and Software Engineering, Hohai University, Nanjing, China

Bin Tang   CSTB@HHU.EDU.CN
Key Laboratory of Water Big Data Technology of Ministry of Water Resources,
College of Computer Science and Software Engineering, Hohai University, Nanjing, China

Baoliu Ye   YEBL@NJU.EDU.CN
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Abstract

Oblique Decision Tree (ODT) partitions the feature space using linear combinations of features, in contrast to a conventional Decision Tree (DT) which is restricted to axis-parallel splits. ODT has been proven to have a stronger representation ability than DT, as it provides a way to create shallower tree structures while still approximating complex decision boundaries. However, its learning efficiency is still insufficient, since the linear projections cannot be transmitted to the child nodes, resulting in a waste of model parameters. In this work, we propose an enhanced ODT method with Feature Concatenation (`FC-ODT`), which enables in-model feature transformation to transmit the projections along the decision paths. Theoretically, we prove that our method enjoys a faster consistency rate w.r.t. the tree depth, indicating that our method possesses a significant advantage in generalization performance, especially for shallow trees. Experiments show that `FC-ODT` outperforms the other decision trees with a limited tree depth.

Keywords: oblique decision tree, feature concatenation, learning theory

1 Introduction

Ensemble learning architectures employing decision trees, notably Random Forests (Breiman, 2001; Geurts et al., 2006) and Gradient Boosted Decision Trees (Friedman, 2001; Chen and

---

✉. Corresponding author

Guestrin, 2016; Ke et al., 2017), have become prevalent solutions for high-dimensional regression/classification problems with inherent ill-posedness (Vershynin, 2018). Their efficacy extends to diverse domains, including causal effect estimation (Wager and Athey, 2018; Doubleday et al., 2022), temporal sequence analysis (Kane et al., 2014), signal reconstruction (Pal, 2005), and computer vision applications (Payet and Todorovic, 2012; Kontschieder et al., 2014). These methods are usually collections of decision trees with axis-aligned splits, such as CART (Breiman et al., 1984) or C4.5 (Quinlan, 1993), that is, the trees only split along feature dimensions, due to their computational efficiency and ease of tuning.

However, the axis-parallel split methods often require very deep trees with complicated step-like decision boundaries when faced with high-dimensional data, leading to increased variance. Therefore, Olique Decision Tree (ODT) (Breiman et al., 1984, Section 5.2) is proposed to use oblique decision boundaries, potentially simplifying the boundary structure. And axis-parallel decision trees can be considered a special case of ODT when the oblique projection direction is only selected from the set of basis vectors. Theoretically, ODT has been proven to have stronger representation capabilities and the potential to achieve better learning properties (Cattaneo et al., 2024).

The major limitations of ODT are *the excessive number of model parameters* in decision paths and *high overfitting risk* in deep nodes (Cattaneo et al., 2024). In fact, the number of parameters required for the learning model to achieve a certain level of performance characterizes the **learning efficiency** of the algorithm. Although variant ODTs (Murthy et al., 1994; Brodley and Utgoff, 1995) use linear combinations of features in each node, they do not transmit projection information to the child nodes. The way of retraining in each node wastes the model parameters we invested in the projection selection, leading to the insufficient learning efficiency of ODT. On the other hand, as the tree grows, the number of samples in deep nodes rapidly decreases. ODT ignores the information obtained from the previous projection selection and retrains the linear model with limited samples, which can lead to severe overfitting risk (Shalev-Shwartz and Ben-David, 2014).

Previous studies often attempt to deal with these limitations via optimization but ignore the impact of wasted projection information. For example, Zhu et al. (2020) utilize the mixed-integer optimization (MIO) strategy to reduce the projection parameters by $\mathbb{L}^1$ regularization. López-Chau et al. (2013) and Tomita et al. (2020) reduce the prediction variance of linear models by introducing $\mathbb{L}^2$ regularization terms or randomization, thus alleviating the overfitting risk. However, these methods can only deal with the learning process in each node separately and cannot consider the relationship between nodes in different layers. To overcome this challenge, we must note that, the transmission of projection information between nodes in different layers plays a critical role.

In this work, we propose an enhanced `Oblique Decision Tree`, `FC-ODT`, that leverages a `Feature Concatenation` mechanism to improve learning efficiency. This mechanism facilitates a layer-by-layer feature transformation during the tree's construction so that the optimized projection information in the node can be transmitted to its child nodes. Meanwhile, the inductive bias brought by the concatenated features combined with the ridge regression method enables the retraining of the linear model in deep nodes to shrink high weights of original features across the linear model per the $\mathbb{L}^2$ penalty term. As a result, `FC-ODT` effectively mitigates multicollinearity by shrinking the coefficients of correlated features

and helps alleviate overfitting by imposing a form of constraint. The contributions are summarized as follows:

- We are the first to establish in-model feature transformation in a single decision tree, dealing with the problem of parameter waste caused by the split strategy of ODT.

- We prove that the consistency rate of `FC-ODT` is faster than traditional ODTs and demonstrate that the feature concatenation mechanism helps improve the learning efficiency of tree construction.

- Experiments on simulated datasets verify the faster consistency rate of `FC-ODT`, and experiments on real-world datasets demonstrate that `FC-ODT` outperforms other SOTA ODTs.

Organization The rest of this article is organized as follows. Section 2 reviews previous work. Section 3 introduces essential background knowledge and notations. Section 4 presents the `FC-ODT` for regression tasks. Section 5 proves the consistency rate of `FC-ODT`. Section 7 conducts simulation and real-world experiments to verify our theoretical results. Section 8 concludes our work with prospects.

2 Related Work

2.1 Oblique Decision Tree

ODT alleviates the problem of high variance of DT in high-dimensional settings, but faces extremely high complexity at each node to find the optimal split and suffers the overfitting risk at the deep node. Heath et al. (1993) and Murthy et al. (1994) propose combining random perturbations with the hill-climbing algorithm to search for optimal split points. Further research has demonstrated the potential of evolutionary computation in ODT training, which employs global optimization strategies to construct decision tree models with enhanced accuracy and structural compactness (Cantú-Paz and Kamath, 2003; Blockeel et al., 2023). It also helps to enhance the individual diversity in the tree ensemble, thereby improving the predictive performance of the ensemble (Zhang et al., 2021; Ferigo et al., 2023). Recently, Bertsimas and Dunn (2017) and Zhu et al. (2020) introduce the MIO strategy to further improve the efficiency of solving projection directions. Unlike these deterministic approximation algorithms, another intriguing and practical research direction involves generating candidate projection vectors via data-driven approaches. Some methods utilize dimensionality reduction techniques, such as PCA (Menze et al., 2011) and LDA (López-Chau et al., 2013). Tomita et al. (2020) show that sparse random projections or random rotations can also be introduced by incorporating. Recently, some studies have extended ODTs to unsupervised learning frameworks such as clustering, demonstrating its advantages in representation ability (Stepišnik and Kocev, 2021; Ganaie et al., 2022). However, the explanation for their success is largely based on heuristics, until Cattaneo et al. (2024) demonstrate the consistency rate of excess risk for individual ODT.

2.2 Feature Concatenation

Deep Forest (Zhou and Feng, 2017) successfully constructs non-differentiable deep models by implementing feature concatenation mechanisms that enable in-model feature transforma-

tion based on decision trees. This mechanism has been theoretically proven to effectively improve the consistency rate of tree-based ensembles (Arnould et al., 2021; Lyu et al., 2022). In addition, feature concatenation also has strong scalability and can adapt to different learning tasks by screening concatenated features. Recent research has expanded the feature concatenation to some specific settings, such as multi-label learning (Yang et al., 2020) and semi-supervised learning (Wang et al., 2020). Although feature concatenation has been widely used in ensemble learning, this work is still the first to introduce it in tree construction.

3 Preliminary

3.1 Setting

Our analysis focuses on a regression framework. The learning process employs a training dataset S_n, containing i.i.d. observations drawn from a joint distribution $\mathbb{P}(\boldsymbol{x}, y) = \mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(\boldsymbol{x})$ over the domain $\mathcal{X} \times \mathcal{Y} \subseteq [0,1]^d \times \mathbb{R}$. The data-generating mechanism follows:

$$y = f(\boldsymbol{x}) + \epsilon \ , \tag{1}$$

where the regression function $f(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}]$ models the conditional expectation of y, while ϵ denotes zero-mean noise with $\mathbb{E}[\epsilon] = 0$ and $\text{Var}[\epsilon] < \sigma^2$.

Our goal is to construct an algorithmic tree estimator $h_n(\cdot, T, S_n)\colon [0,1]^d \to \mathbb{R}$. Here, T represents a tree architecture induced by S_n. We adopt the abbreviated notation $h_{T,n}(\boldsymbol{x}) = h_n(\boldsymbol{x}, T, S_n)$. The performance of $h_{T,n}$ is quantified by its mean squared error:

$$R(h_{T,n}) = \mathbb{E}\left[(h_{T,n}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\right] \ , \tag{2}$$

with the expectation computed over $\boldsymbol{x}$ given a fixed training set S_n. As the sample size $n \to \infty$, a sequence of estimators $\{h_{T,i}\}_{i=1}^n$ emerges. Such a sequence is consistent iff $R(h_{T,n}) \to 0$.

3.2 Decision Tree

A decision tree is a data structure commonly constructed through recursive binary partitioning, often implemented in a top-down hierarchical manner using greedy optimization. Alternative strategies, such as bottom-up pruning or global optimization frameworks, may also be employed depending on computational constraints. Under the CART framework (Breiman et al., 1984), splitting operation partitions a parent node t (representing a region $\mathcal{X}$) into t_L and t_R, achieving maximal impurity reduction measured by mean squared error (MSE).

$$\widehat{\Delta}(b, \boldsymbol{a}, t) = \tfrac{1}{n} \sum_{\boldsymbol{x} \in t} (y_i - \bar{y}_t)^2 - \tfrac{1}{n} \sum_{\boldsymbol{x} \in t} \left(y_i - \bar{y}_{t_L}\mathbb{1}(\boldsymbol{a}^\top \boldsymbol{x}_i \leq b) - \bar{y}_{t_R}\mathbb{1}(\boldsymbol{a}^\top \boldsymbol{x}_i > b)\right)^2, \tag{3}$$

for the pair $(b, \boldsymbol{a})$, $\mathbb{1}(\cdot)$ represents the indicator function, and $\bar{y}_t$ is defined as the sample mean of y_i observations for which the associated $\boldsymbol{x}_i$ data falls within node t.

For traditional DT such as CART (Breiman et al., 1984, Section 2.2), splits always follow the direction parallel to the axis. In this case, the feature space partition learned by DT is always a set of hyper-rectangles. When encountering high-dimensional data, deep

trees with high complexity are often generated, leading to overfitting risks. ODT such as oblique CART uses linear combinations between features as the basis for partitioning, thus expanding the hypothesis set of projection direction $\boldsymbol{a}$ to $\mathbb{R}^d$.

In Eqn. (2), the conditional response output by the decision tree T as an estimator is the average of the target values of all training samples in the leaf node,

$$h_{T,n}(\boldsymbol{x}) = \bar{y}_t = \frac{1}{n(t)} \sum_{\boldsymbol{x}_i \in t} y_i \ , \tag{4}$$

where $n(t)$ characterizes the local sample size within node t.

The existing work to improve ODT mainly focuses on the computational efficiency and regularization of Eqn. (3), which often only directly affects the splitting of nodes in each layer and cannot deal with the relationship between nodes in different layers. The average response in Eqn. (4) wastes the computational cost invested in projection selection. The existing ODT learning frameworks are unable to effectively transmit the projection information of the parent node to its child nodes, resulting in limited learning properties.

4 The Proposed Approach

This section presents FC-ODT, whose key idea is to introduce a feature concatenation mechanism so that the layer-by-layer splitting process can achieve in-model feature transformation just like neural networks. FC-ODT can be not only practical but also a heuristic algorithm with provably better learning properties. It consists of three steps: feature concatenation, finding oblique splits, and tree construction, which are detailed in Algorithms 1-3.
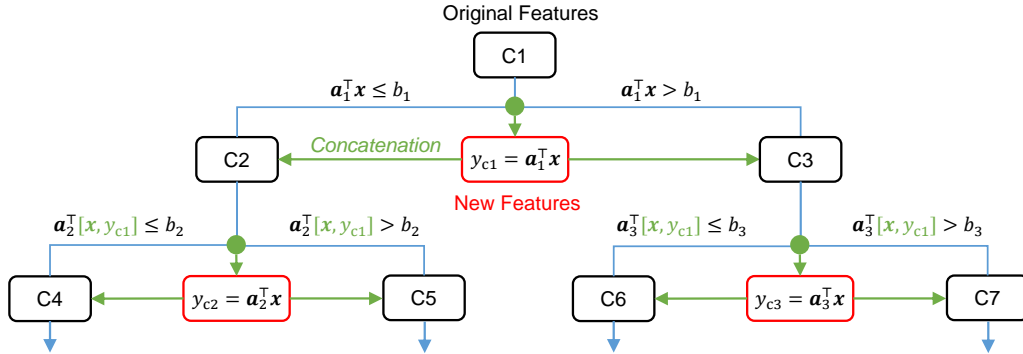

Figure 1: Illustration of our FC-ODT framework, where $[x, y]$ denotes the feature concatenation between x and y.

4.1 Feature Concatenation

As shown in Figure 1, when the parent node t of each level splits, it will learn the oblique decision rule $\boldsymbol{a}^\top \boldsymbol{x} < b$. We will record the projected score $\tilde{y}_t = \boldsymbol{a}^\top \boldsymbol{x}$ as the augmented

---
Algorithm 1 Feature concatenation.

Input: The data $\boldsymbol{X}$ falling into the current internal node t of a tree; $\tilde{\boldsymbol{y}}_t$ denotes the generated new feature for the data of node t.

Output: The data matrix $\boldsymbol{X}_t$ after feature concatenation.

1: **function** $\boldsymbol{X}_t \leftarrow \text{FEACONC}(\boldsymbol{X}, \tilde{\boldsymbol{y}}_t)$
2:      $\boldsymbol{X}_t \leftarrow [\boldsymbol{X}, \tilde{\boldsymbol{y}}_t]$
3:      **return** $\boldsymbol{X}_t$
4: **end function**

Algorithm 2 Finding the oblique split in an internal node.

Input: The training set $S^{(t)} = (\boldsymbol{X}, \boldsymbol{y})$ falling into the internal node t of a decision tree; λ is the hyper-parameter controlling the regularization strength.

Output: The projection $\boldsymbol{a}^*$ and the threshold b^*

1: **function** $(\boldsymbol{a}^*, b^*) \leftarrow \text{FINDOBLIQUESPLIT}(S^{(t)})$
2:      $\boldsymbol{a}^* \leftarrow \arg\min_{\boldsymbol{a}} \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{a}\|_2^2 + \lambda \|\boldsymbol{a}\|_2^2$
3:      $b^* \leftarrow \arg\max_b \widehat{\Delta}(b, \boldsymbol{a}^*, t)$
4:      **return** $(\boldsymbol{a}^*, b^*)$
5: **end function**

---

feature after entering the child node, and concatenate it with the original feature $\boldsymbol{x}$ of the sample in the parent node to obtain the complete feature of the sample in the child node

$$\boldsymbol{X}_t \leftarrow [\boldsymbol{X}, \tilde{\boldsymbol{y}}_t] \;, \tag{5}$$

where $\boldsymbol{X}_t \in [0, 1]^{n \times d}$ denotes the feature matrix of samples in node t, and $[\cdot, \cdot]$ denotes the concatenation of two feature matrices to form a new feature matrix. We encode the optimized projection information from the parent node into the new feature, and pass it along the decision path to the child nodes through concatenation operations to reduce the computational cost. This idea is similar to the Boosting algorithm (Bartlett et al., 1998), which uses different learners to process independent parts of data separately, and a stronger learner can be obtained by the ensemble. Feature concatenation has been used in deep forests to construct boosting frameworks to improve learning efficiency (Lyu et al., 2019). This feature concatenation process is repeated for each split until the desired tree size and constructs a `FC-ODT`, which is detailed in Algorithm 1.

4.2 Finding Oblique Splits

For `FC-ODT`, we rely on linear combinations of multiple features for binary splits at each node. For a sample $\boldsymbol{x} \in \mathbb{R}^d$ in a feature space, the decision function at the node t can be formulated as:

$$\boldsymbol{a}_t^\top \boldsymbol{x} < b_t \;, \tag{6}$$

with coefficients $\boldsymbol{a}_t$ denoting the projection direction and b_t indicating the splitting threshold. Determining the optimal value for $\boldsymbol{a}_t$ is proved to be more challenging than identifying a single optimal feature and setting a suitable threshold for a split in the univariate scenario (Murthy et al., 1994).

Considering that we have introduced a feature concatenation mechanism in the ODT generation process, this can lead to collinearity issues between the original and augmented features of the sample. Especially, the correlation between the augmented feature $\boldsymbol{a}^\top \boldsymbol{x}$ and the label y is high, which can cause overfitting problems during the training process as the augmented feature has a significant impact on the model's output. Therefore, we choose ridge regression to find the projection direction of the split

$$\boldsymbol{a}_t(\lambda) = \arg\max_{\boldsymbol{a}_t} \|\boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{a}_t\|_2^2 + \lambda \|\boldsymbol{a}_t\|_2^2 \;, \tag{7}$$

using regularization parameter λ.

Algorithm 3 Learning an oblique decision tree with feature concatenation (`FC-ODT`).

Input: $S_n = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ is the training set; $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ is the feature matrix; Γ is a set of split eligibility criteria; pre-specified number of leaves t_n

Output: `FC-ODT` regression tree T

1: **function** $T \leftarrow \text{TREECONSTRUCTION}(S_n)$
2:      $t \leftarrow 1$
3:      $t_c \leftarrow 1$
4:      $S^{(t)} \leftarrow S_n$
5:      **while** $t < t_c + 1$ **do**
6:          **if** Γ satisfied **then**
7:              $(\boldsymbol{a}^*, b^*) \leftarrow \text{FINDOBLIQUESPLIT}(S^{(t)})$ $\triangleright$ refer to Algorithm 2
8:              $\tilde{\boldsymbol{y}}_t \leftarrow \boldsymbol{X}_t^\top \boldsymbol{a}^*$ $\triangleright$ generate the new feature via linear projection
9:              $\boldsymbol{X}_t \leftarrow \text{FEACONC}(\boldsymbol{X}, \tilde{\boldsymbol{y}}_t)$ $\triangleright$ update $\boldsymbol{X}_t$, refer to Algorithm 1
10:             $\boldsymbol{y}_t \leftarrow \boldsymbol{y}_t - \tilde{\boldsymbol{y}}_t$ $\triangleright$ update $\boldsymbol{y}_t$
11:             $S^{(K+1)} \leftarrow \{i \colon \boldsymbol{x}_i^\top \boldsymbol{a}^* < b^*, \quad \forall i \in S^{(t)}\}$
12:             $S^{(K+2)} \leftarrow \{i \colon \boldsymbol{x}_i^\top \boldsymbol{a}^* \geq b^*, \quad \forall i \in S^{(t)}\}$
13:             $\boldsymbol{a}_t \leftarrow \boldsymbol{a}^*$
14:             $b_t \leftarrow b^*$
15:             $\kappa_t \leftarrow \{t_c + 1, t_c + 2\}$
16:             $t_c = t_c + 2$
17:          **else**
18:             $(\boldsymbol{a}_t, b_t, \kappa_t) \leftarrow \text{NULL}$
19:          **end if**
20:          $t \leftarrow t + 1$
21:      **end while**
22:      **return** $\left(S^{(1)}, \{\boldsymbol{a}_t, b_t, \kappa_t, \tilde{\boldsymbol{y}}_t\}_{t=1}^{t_n}\right)$
23: **end function**

With this choice, the node model is optimized as (Smith and Campbell, 1980)

$$\boldsymbol{a}_t(\lambda) = \arg\max_{\|\boldsymbol{a}_t\|=1} \text{Cov}(\boldsymbol{X}^\top \boldsymbol{a}_t, \boldsymbol{y}) \cdot \frac{\text{Var}(\boldsymbol{X}^\top \boldsymbol{a}_t)}{\text{Var}(\boldsymbol{X}^\top \boldsymbol{a}_t) + \lambda} \;. \tag{8}$$

Ridge regression can shrink high feature weights across the linear model per the $\mathbb{L}^2$ penalty term. This reduces the complexity of the model and helps make model predictions less erratically dependent on any one or more features.

4.3 Tree Construction

With these two steps at hand, we can iteratively perform node splitting and feature concatenation to construct an enhanced oblique decision tree with the ability of in-model feature transformation, named `FC-ODT`. The tree construction procedure is described in detail in Algorithm 3.

To prepare for analyzing the convergence rate of excess risk in Section 5, here we introduce the following definition.

Definition 1 (Orthonormal decision stumps). *The orthonormal decision stumps is defined as*

$$\psi_t(\boldsymbol{x}) = \frac{\boldsymbol{c}_{t_L}(\boldsymbol{x})n(t_R) - \boldsymbol{c}_{t_R}(\boldsymbol{x})n(t_L)}{\sqrt{w(t)n(t_L)n(t_R)}} \ , \tag{9}$$

where $\boldsymbol{c}_{t_L}(\boldsymbol{x}) = \boldsymbol{x}^\top \left(\boldsymbol{X}_{t_L}^\top \boldsymbol{X}_{t_L} + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{X}_{t_L}^\top$ *and* $\boldsymbol{X}_{t_L} = \boldsymbol{X} \circ \mathbb{1}(\boldsymbol{x} \in t_L)$*, where* $\circ$ *denotes Hadamard multiplication. For internal nodes* $t \in [T]$*, the equality* $w(t) = n(t)/n$ *represents the proportion of observations assigned to node* t*.*

Next, we demonstrate that tree estimation is the empirical orthogonal projection of $\boldsymbol{y}$ onto the linear span of orthonormal decision stumps.

Lemma 2 (Orthogonal tree expansion). *If T denotes a decision tree constructed by* `FC-ODT` *method, then its output* (4) *admits the following orthogonal expansion*

$$h_{T,n}(\boldsymbol{x}) = \sum_{t \in [T]} \langle \boldsymbol{y}, \boldsymbol{\psi}_t \rangle_n \cdot \psi_t(\boldsymbol{x}) \ , \tag{10}$$

where $\boldsymbol{\psi}_t = (\psi_t(\boldsymbol{x}_1), \ldots, \psi_t(\boldsymbol{x}_n))^\top$ *is defined in Definition 1. Let $h_{T,n}$ be the empirical orthogonal projection of $\boldsymbol{y}$ onto the linear span of $\{\boldsymbol{\psi}_t\}_{t \in [T]}$. Then, we have $|\langle \boldsymbol{y}, \boldsymbol{\psi}_t \rangle|^2 = \widehat{\Delta}(\hat{b}, \hat{\boldsymbol{a}}, t)$.*

Remark 3. *Diverging from conventional ODT's orthogonal cut paradigm, which myopically partitions feature space and optimizes locally constant predictors at each bifurcation (Cattaneo et al., 2024), Lemma 2 shows that* `FC-ODT` *uses a feature concatenation mechanism to make the prediction factors of its orthogonal decision stumps contain information about the projection selection. Especially, the inductive bias brought by new features leads to child nodes tending to search for residual ridge regression solutions near the projection of the parent node, which enables the child nodes to learn local linear structures more efficiently in the subspaces.*

Representation of feature concatenation Our method passes the linear projection information in the parent node to the child nodes through feature concatenation. Due to the linear transferability, it can be known that the method does not change the linear nature of the node splits in the ODT. In other words, an ODT obtained using feature concatenation can be completely represented by a traditional ODT. This also corresponds to what we emphasized in the abstract, that the goal of this paper is to show that feature concatenation improves the learning efficiency of ODTs, rather than changing their representation.

Time complexity In terms of time complexity, a single node in FC-ODT needs to solve a $d+1$-dimensional ridge regression problem to get the optimal projection direction $\boldsymbol{a}^*$ and threshold b^*, so its complexity is $O(n(d+1)^2+(d+1)^3) = O(d^3+(n+3)d^2)$. Since this paper uses an oblique decision tree with restricted maximum depth, the computational complexity of the overall tree is $O((2^K-1)(n(d+1)^2+(d+1)^3)) = O((2^K-1)(d^3+(n+3)d^2))$. This is of the same order as the computational complexity $O((2^K-1)(d^3+nd^2))$ of Ridge-ODT. Analysis of this computational complexity reveals that the magnitude of complexity is dominated by $O(nd^2)$ when the amount of data is much larger than the number of feature dimensions, and vice versa, when the number of feature dimensions is much larger than the amount of data, the magnitude of complexity is dominated by $O(d^3)$. Here, the dependence of computational complexity on the number of feature dimensions can be reduced by introducing an evolutionary algorithm, but it also introduces an additional influencing factor of the number of iterations. In practice, evolutionary algorithms are indeed more advantageous in terms of computational efficiency. However, evolutionary algorithms often use heuristic ideas, and their theoretical performance is difficult to portray, so it is not possible to compare them with our method in terms of learning efficiency theoretically.

5 Theoretical Analysis

In this section, we show that `FC-ODT` can achieve a faster convergence rate of consistency with respect to the tree depth K. Detailed proofs can be found in the supplementary. First, we introduce the definition of an additive model, which is a common family of functions in statistical analysis.

Definition 4 (Ridge expansions (Cattaneo et al., 2024)). *Consider the family of functions consisting of finite linear combinations of ridge functions:*

$$\mathcal{G} = \left\{ g(\boldsymbol{x}) = \sum_{k=1}^K g_k(\boldsymbol{a}_k^\top \boldsymbol{x}),\ \boldsymbol{a}_k \in \mathbb{R}^d, g_k \colon \mathbb{R} \to \mathbb{R},\quad k = 1,\dots,K,\ \|g\|_{\mathcal{L}_1} < \infty \right\}, \quad (11)$$

where $\|g\|_{\mathcal{L}_1}$ is a total variation norm defined in Definition 5.

Definition 5 (Total variation norm in node t). *Let $V(h, \boldsymbol{a}, t)$ quantify the total variation of ridge function $\boldsymbol{x} \to h(\boldsymbol{a}^\top \boldsymbol{x})$ over node t via*

$$V(h, \boldsymbol{a}, t) = \sup_{\mathcal{P}} \sum_{\ell=0}^{|\mathcal{P}|-1} |h(z_{\ell+1} - h(z_\ell))| , \quad (12)$$

where the supremum considers all finite partitions $\mathcal{P} = \{z_0, \dots, z_{|\mathcal{P}|}\}$ of the projected interval $I(\boldsymbol{a}, t) = [\min_{\boldsymbol{x} \in t} \boldsymbol{a}^\top \boldsymbol{x}, \max_{\boldsymbol{x} \in t} \boldsymbol{a}^\top \boldsymbol{x}]$. For any function $f \in \mathcal{F} = \mathrm{cl}(\mathcal{G})$, the $\mathcal{L}_1$-total variation norm at node t is characterized as:

$$\|f\|_{\mathcal{L}_1(t)} \triangleq \liminf_{\epsilon \downarrow 0}\ _{g \in \mathcal{G}} \left\{ \sum_{k=1}^K V(g_k, \boldsymbol{a}_k, t) \colon g(\boldsymbol{x}) = \sum_{k=1}^K g_k(\boldsymbol{a}_k^\top \boldsymbol{x}),\ \|f - g\| \le \epsilon \right\}. \quad (13)$$

The models that decompose the regression function into a sum of ridge functions have been widely recognized and promoted by Stone (1985), as well as Hastie and Tibshirani

9

(1987). In particular, the consistency of ODT under this assumption has been proven by Cattaneo et al. (2024). Within this analytical framework, we investigate how feature-space augmentation in `FC-ODT` governs model behavior. Our theoretical derivations necessitate the subsequent stochastic data assumption:

[Exponential tails (Cattaneo et al., 2024)] The conditional distribution of y given $\boldsymbol{x}$ has exponentially decaying tails. That is, there exist positive constants c_1, c_2, and M, such that for all $\boldsymbol{x} \in \mathcal{X}$,

$$\mathbb{P}(|y| > B + M \mid \boldsymbol{x}) \leq c_1 \exp(-c_2 B^2), \ B \geq 0 \ . \tag{14}$$

Theorem 6 (Consistency rate for `FC-ODT`). *Let the conditional expectation $f(\boldsymbol{x})$ be from the ridge functions defined by Definition 4 and the conditional distribution $\mathbb{P}(y|\boldsymbol{x})$ satisfying Assumption 5. Consider a training set of n samples drawn from this distribution and a K-layer decision tree T_K constructed by `FC-ODT` on the training set. Then, for any $K \geq 1$ and $n \geq 1$, we have*

$$\mathbb{E}\left[\|h_{T_K,n}(\boldsymbol{x}) - f(\boldsymbol{x})\|_2^2\right] \leq 2\inf_{f \in \mathcal{F}}\left\{\|g - f\|_2^2 + C_1 \frac{\|g\|_{\mathcal{L}_1}^2}{K^2} + C_2 \frac{2^K d \log^2 n}{n}\right\}, \tag{15}$$

where $C_1 = C_1(B, M)$ and $C_2 = C_2(c_1.c_2, B, M)$ are two positive constants.

Remark 7. *Theorem 6 proves that `FC-ODT` is consistent, and the convergence rate is the same as conventional ODTs with the increase of sample size n, both of which are better than axis-parallel DT. Since the feature concatenation mechanism transmits the projection information to child nodes, the convergence rate of excess risk is $\mathcal{O}(1/K^2)$ w.r.t. tree depth K, which is faster than conventional ODTs with order $\mathcal{O}(1/K)$ in (Cattaneo et al., 2024). This result demonstrates that `FC-ODT` has advantages in learning efficiency compared to conventional ODTs. Especially considering the high computational complexity of optimal linear projection and the fact that deep decision paths can impair the interpretability of the model, we often limit the tree depth of ODT in practice (Zhu et al., 2020). When K is small, the theoretical advantages of `FC-ODT` over ODT become more significant.*

6 Proofs

In this section, we provide the detailed proofs for the main theorem and lemma.

6.1 Proof of Lemma 2

Proof Set $\mathcal{U}_t = \{u(\boldsymbol{x})\mathbb{1}(\boldsymbol{x} \in t_L) + v(\boldsymbol{x})\mathbb{1}(\boldsymbol{x} \in t_R) : u, v \in \text{span}(\mathcal{H})\}$ and consider the closed subspace $\mathcal{V}_t = \{v(\boldsymbol{x})\mathbb{1}(\boldsymbol{x} \in t) : v \in \text{span}(\mathcal{H})\}$. By the orthogonal decomposition property of Hilbert spaces, we can express $\mathcal{U}_t$ as the direct sum $\mathcal{V}_t \oplus \mathcal{V}_t^{\perp}$, where $\mathcal{V}_t^{\perp} = \{u \in \mathcal{U}_t : \langle u, v\rangle_n = 0, \forall v \in \mathcal{V}_t\}$. Let Ψ_t be any orthonormal basis for $\mathcal{V}_t$ that includes $w^{-1/2}(t)\mathbb{1}(\boldsymbol{x} \in t)$, where $w(t) = n(t)/n$. Let $\Psi_t^{\perp}$ be any orthonormal basis for $\mathcal{V}_t^{\perp}$ that includes the decision stump defined by Eqn. (9).

Consider that $\tilde{y}_t(\boldsymbol{x}) = \hat{\boldsymbol{a}}_t^{\top}\boldsymbol{x}$ is the projection of $\boldsymbol{y}$ onto $\mathcal{V}_t$, where $\hat{\boldsymbol{a}}_t = \left(\boldsymbol{X}_{t_L}^{\top}\boldsymbol{X}_{t_L} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}_{t_L}^{\top}\boldsymbol{y}$, we have

$$\tilde{y}_t(\boldsymbol{x}) = \sum_{\psi \in \Psi_t} \langle \boldsymbol{y}, \boldsymbol{\psi}\rangle_n \psi(\boldsymbol{x}) \ , \tag{16}$$

and

$$\tilde{y}_{t_L}(\boldsymbol{x})\mathbb{1}(\boldsymbol{x} \in t_L) + \tilde{y}_{t_R}(\boldsymbol{x})\mathbb{1}(\boldsymbol{x} \in t_R)$$
$$= \sum_{\psi \in \Psi_t \cup \Psi_t^{\perp}} \langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n \psi(\boldsymbol{x}) . \tag{17}$$

Using the above expansions, observe that for each internal node t,

$$\sum_{\psi \in \Psi_t^{\perp}} \langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n \psi(\boldsymbol{x}) = (\tilde{y}_{t_L} - \tilde{y}_t)\mathbb{1}(\boldsymbol{x} \in t_L) + (\tilde{y}_{t_R} - \tilde{y}_t)\mathbb{1}(\boldsymbol{x} \in t_R) . \tag{18}$$

For each $\boldsymbol{x} \in \mathcal{X}$, let $t_0, t_1, \ldots, t_{K-1}, t_K = t$ be the unique path from the root node t_0 to the terminal node t that contains $\boldsymbol{x}$. Next, sum (18) over all internal nodes and telescope the successive internal node outputs to obtain

$$\sum_{k=0}^{K-1} (\tilde{y}_{t_{k+1}}(\boldsymbol{x}) - \tilde{y}_{t_k}(\boldsymbol{x})) = \tilde{y}_{t_K}(\boldsymbol{x}) - \tilde{y}_{t_0}(\boldsymbol{x}) = \tilde{y}_t(\boldsymbol{x}) - \tilde{y}(\boldsymbol{x}) , \tag{19}$$

where $\tilde{y}$ is the linear estimation output by solving ridge regression in the root node:

$$\min_{\boldsymbol{a}} \|\boldsymbol{y} - \boldsymbol{X}^{\top}\boldsymbol{a}\|_2^2 + \lambda\|\boldsymbol{a}\|_2^2 . \tag{20}$$

Combining Eqns. (18) and (19), we have

$$\sum_{t \in [T]} \tilde{y}_t(\boldsymbol{x})\mathbb{1}(\boldsymbol{x} \in t) = \tilde{y} + \sum_{t \in [T]\setminus\{t_0\}} \sum_{\psi \in \Psi_t^{\perp}} \langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n \psi(\boldsymbol{x})$$
$$= \sum_{t \in [T]} \sum_{\psi \in \Psi_t^{\perp}} \langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n \psi(\boldsymbol{x}) , \tag{21}$$

where we recall that the root node t_0 is an internal node of T.

Finally, the decrease in impurity identity (3) satisfies that:

$$\widehat{\Delta}(\hat{b}, \hat{\boldsymbol{a}}, t) = \frac{1}{n} \sum_{\boldsymbol{x}_i \in t} (y_i - \tilde{y}_t(\boldsymbol{x}))^2$$
$$- \frac{1}{n} \sum_{\boldsymbol{x}_i \in t} (y_i - \tilde{y}_{t_L}(\boldsymbol{x}_i)\mathbb{1}(\boldsymbol{x}_i \in t_L) - \tilde{y}_{t_R}(\boldsymbol{x}_i)\mathbb{1}(\boldsymbol{x}_i \in t_R))^2$$
$$= \left(\frac{1}{n} \sum_{\boldsymbol{x}_i \in t} y_i^2 - \sum_{\psi \in \Psi_t} |\langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n|^2 \right) - \left(\frac{1}{n} \sum_{\boldsymbol{x}_i \in t} y_i^2 - \sum_{\psi \in \Psi_t \cup \Psi_t^{\perp}} |\langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n|^2 \right) \tag{22}$$
$$= \sum_{\psi \in \Psi_t^{\perp}} |\langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n|^2 .$$

$\blacksquare$

6.2 Proof of Theorem 6

Proof Following the proofs in (Cattaneo et al., 2024), we begin by splitting the MSE (averaging only with respect to the joint distribution of $\{\mathcal{A}_t \colon t \in [T_k]\}$) into two terms

$$\mathbb{E}_{T_k}[\|h_{T_k,n}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2] = E_1 + E_2 , \tag{23}$$

where

$$\begin{aligned}
E_1 =& \mathbb{E}_{T_k}[\|h_{T_k,n}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2] - 2(\mathbb{E}_{T_k}[\|h_{T_k,n}(\boldsymbol{x}) - \boldsymbol{y}\|_n^2 - \|\boldsymbol{y} - f(\boldsymbol{x})\|_n^2]) \\
& - \alpha(n,k) - \beta(n) , \tag{24} \\
E_2 =& 2(\mathbb{E}_{T_k}[\|h_{T_k,n}(\boldsymbol{x}) - \boldsymbol{y}\|_n^2] - \|\boldsymbol{y} - f(\boldsymbol{x})\|_n^2) + \alpha(n,k) + \beta(n) , \tag{25}
\end{aligned}$$

and $\alpha(n,k)$ and $\beta(n)$ are positive sequences that will be specified later.

To bound $\mathbb{E}[E_1]$, we split our analysis into two cases based on the observed data y_i. Accordingly, we have

$$\mathbb{E}[E_1] = \mathbb{E}[E_1 \mathbb{1}(\forall i \colon |y_i| \leq B)] + \mathbb{E}[E_1 \mathbb{1}(i \colon |y_i| > B)] \tag{26}$$

Firstly, we deal with the bounded term $\mathbb{E}[E_1 \mathbb{1}(\forall i \colon |y_i| \leq B)]$. According to (Cattaneo et al., 2024, pages 24-25) and (Györfi et al., 2002, Lemma 13.1, Theorem 11.4), let $R = QB$ such that $R \geq M \geq \|\boldsymbol{y}\|_\infty$, we have

$$\begin{aligned}
& \mathbb{P}\left(\mathbb{E}_{T_K}[\|h_{T_k,n}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2] \geq E_2, \ \forall i \colon |y_i| < B\right) \\
& \leq 14 \sup_{\boldsymbol{x}^n} \mathcal{N}\left(\frac{\beta(n)}{40R}, \mathcal{F}_{n,k}(R), \mathcal{L}_1(\mathbb{P}_{\boldsymbol{x}^n})\right) \exp\left(-\frac{\alpha(n,k)n}{2568R^4}\right) , \tag{27}
\end{aligned}$$

where $\mathcal{N}\left(\frac{\beta(n)}{40R}, \mathcal{F}_{n,k}(R), \mathcal{L}_1(\mathbb{P}_{\boldsymbol{x}^n})\right)$ denotes the covering number of $\mathcal{F}_{n,k}(R)$ by balls of radius $r > 0$ in $\mathcal{L}_1(\mathbb{P}_{\boldsymbol{x}^n})$ with respect to the empirical discrete measure $\mathbb{P}_{\boldsymbol{x}^n}$ on $\boldsymbol{x}^n$ which satisfies that

$$\mathcal{N}\left(\frac{\beta(n)}{40R}, \mathcal{F}_{n,k}(R), \mathcal{L}_1(\mathbb{P}_{\boldsymbol{x}^n})\right) \leq \left(3\left(\frac{enp}{d}\right)^d\right)^{2^k} \left(\frac{240eR^2}{\beta(n)}\right)^{\mathrm{VC}(\mathcal{H})2^{k+1}} . \tag{28}$$

Combining Eqns. (27) and (28), we have

$$\mathbb{P}(E_1 \geq 0, \ \forall i \colon |y_i| \leq B) \leq 42 \left(\left(\frac{enp}{d}\right)^d\right)^{2^k} \left(\frac{240eR^2}{\beta(n)}\right)^{\mathrm{VC}(\mathcal{H})2^{k+1}} e^{-\frac{\alpha(n,k)n}{2568R^4}} , \tag{29}$$

so that $\mathbb{E}_1 \geq 0, \forall i \colon |y_i| \leq B \leq 1/n^2$ and $E_1 \mathbb{1}(\forall i \colon |y_i| \leq B) \leq 12R^2$.

Then, by choosing

$$\begin{aligned}
\alpha(n,k) =& \frac{2568R^4 \left(2^k d \log(enp/d) + 2^k \log(3) + \mathrm{VC}(\mathcal{H})2^{k+1} \log\left(\frac{240eR^2}{\beta(n)}\right) + \log\left(14n^2\right)\right)}{n} \\
\beta(n) =& \frac{240eR^2}{n^2} ,
\end{aligned} \tag{30}$$

12

we have

$$\mathbb{E}[E_1 \mathbb{1}(\forall i\colon |y_i| \leq B)] \leq 12R^2 \mathbb{P}(E_1 \geq 0, \ \forall i\colon |y_i| \leq B) \leq \frac{12R^2}{n^2} \leq \frac{12Q^2 B^2}{n^2} \ . \qquad (31)$$

Secondly, for the unbounded term $\mathbb{E}[E_1 \mathbb{1}(i\colon |y_i| > B)]$, by Cattaneo et al. (2024), we have

$$\begin{aligned} &\mathbb{E}[\|h_{T_k,n}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2 \mathbb{1}(i\colon |y_i| > B)] \\ \leq &(Q+1)^2 \sqrt{(n+1)\mathbb{E}[y^4]} \sqrt{nc_1 \exp(-c_2(B-M)^2)} \ . \end{aligned} \qquad (32)$$

and

$$\mathbb{E}[E_2] = \|f - g\|^2 + \mathbb{E}[\|h_{T_k,n}(\boldsymbol{x}) - \boldsymbol{y}\|_n^2 - \|\boldsymbol{y} - g(\boldsymbol{x})\|_n^2] + \alpha(n,k) + \beta(n) \ . \qquad (33)$$

Since the excess risk can be decomposed by the Ridge expansions $g(\boldsymbol{x})$

$$\mathbb{E}[\|h_{T_k,n}(\boldsymbol{x}) - \boldsymbol{y}\|_n^2 - \|\boldsymbol{y} - f(\boldsymbol{x})\|_n^2] = \|f - g\|^2 + \mathbb{E}[\|h_{T_k,n}(\boldsymbol{x}) - \boldsymbol{y}\|_n^2 - \|\boldsymbol{y} - g(\boldsymbol{x})\|_n^2] \ , \qquad (34)$$

we have

$$\begin{aligned} \mathbb{E}_{T_k}[\|h_{T_k,n}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2] \leq &\|f - g\|^2 + \mathbb{E}[\|h_{T_k,n}(\boldsymbol{x}) - \boldsymbol{y}\|_n^2 - \|\boldsymbol{y} - g(\boldsymbol{x})\|_n^2] \\ &+ C_2 \frac{2^K(d + \mathrm{VC}(\mathcal{H})) \log^2 n}{n} \ , \end{aligned} \qquad (35)$$

for some positive constant $C_2(c_1, c_2, B, M)$.

Define the squared node-wise norm and node-wise inner product as $\|\boldsymbol{f}\|_t^2 = \frac{1}{n(t)} \sum_{\boldsymbol{x}_i \in t}(f(\boldsymbol{x}_i))^2$ and $\langle \boldsymbol{f}, \boldsymbol{g} \rangle_t = \frac{1}{n(t)} f(\boldsymbol{x}_i) g(\boldsymbol{x}_i)$. We define the node-wise excess training error as

$$R_K(t) = \|h_{T_K,n}(\boldsymbol{x}) - \boldsymbol{y}\|_t^2 - \|\boldsymbol{y} - \boldsymbol{g}\|_t^2 \ . \qquad (36)$$

Then, we define the total excess training error as:

$$R_K = \sum_{t \in T_K} w(t) R_K(t), \quad w(t) = n(t)/n \ , \qquad (37)$$

where $t \in T_K$ means t is a terminal node of T_K.

According to the orthogonal decomposition of the FC-ODT in Lemma 2, we have

$$R_K = R_{K-1} - \sum_{t \in T_{K-1}} \sum_{\psi \in \Psi_t^\perp} |\langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n|^2 \ . \qquad (38)$$

We denote by $\mathbb{E}_{T_K}$ the expectation is taken with respect to the joint distribution of $\{\mathcal{A}_t\colon t \in [T_K]\}$, conditional on the data. By the definition of R_K, we have

$$\mathbb{E}[R_K] = \mathbb{E}[\|h_{T_K,n}(\boldsymbol{x}) - \boldsymbol{y}\|_n^2 - \|\boldsymbol{y} - \boldsymbol{g}\|_n^2] \geq 0 \ . \qquad (39)$$

Using the law of iterated expectations and the recursive relationship obtained in Eqn. (38), we have

$$\begin{aligned} \mathbb{E}_{T_K}[R_K] &= \mathbb{E}_{T_K}[\mathbb{E}_{T_K|T_{K-1}}[R_K]] \\ &= \mathbb{E}_{T_{K-1}}[R_{K-1}] - \mathbb{E}_{T_{K-1}}\left[\mathbb{E}_{T_K|T_{K-1}}\left[\sum_{t \in T_{K-1}} \sum_{\psi \in \Psi_t^\perp} |\langle \boldsymbol{y}, \boldsymbol{\psi} \rangle_n|^2\right]\right] \ . \end{aligned} \qquad (40)$$

13

According to the sub-optimal probability defined by Cattaneo et al. (2024, Section 2.2) and the sum of the iterative equality (40), we have

$$\mathbb{E}_{T_K}[R_K] = \sum_{t \in T_{K-1}:\, R_{K-1}(t)>0} P_{\mathcal{A}_t}(\kappa) \max_{(b,va)\in\mathbb{R}^{p+1}} \widehat{\Delta}(b, \boldsymbol{a}, t) , \tag{41}$$

where $P_{\mathcal{A}_t}(\kappa) = \mathbb{P}_{\mathcal{A}_t}\left(\max_{(b,\boldsymbol{a})\in\mathbb{R}\times\mathcal{A}_t} \widehat{\Delta}(b, \boldsymbol{a}, t) \geq \kappa \max_{(b,\boldsymbol{a})\in\mathbb{R}^{1+d}} \widehat{\Delta}(b, \boldsymbol{a}, t)\right)$ is defined to quantify the sub-optimality of the learning algorithm theoretically, and $\mathbb{P}_{\mathcal{A}_t}$ denotes the probability w.r.t. the randomness in the learning algorithm $\mathcal{A}_t$.

By (Cattaneo et al., 2024, Lemma 4 and Lemma 6), we have

$$\mathbb{E}_{T_K}[R_K] \leq \mathbb{E}_{T_{K-1}}[R_{K-1}] - \kappa \mathbb{E}_{T_{K-1}} \left[\frac{\mathbb{E}^2_{T_{K-1}}[R_{K-1}]/w(t)}{\sum_{t\in T_{K-1}} Q\|g\|^2_{\mathcal{L}_1(t)}} \right] . \tag{42}$$

Due to the feature concatenation mechanism transmitting projection direction information within the model, the variation $\|g\|^2_{\mathcal{L}_1(t)}$ in the node t in the path decreases with increasing depth. This phenomenon is similar to the decay of residuals in boosting algorithms, and we have obtained a recursion for $\mathbb{E}[R_K]$, by (Cattaneo et al., 2024, Lemma 5), we have

$$\mathbb{E}[R_K] \leq \frac{1}{\kappa \sum_{k=1}^{K} 1/\mathbb{E}[\sum_{t\in T_{k-1}} w(t)Q\|g\|^2_{\mathcal{L}_1(t)}]} , \tag{43}$$

where

$$\mathbb{E}[\sum_{t\in T_{k-1}} w(t)Q\|g\|^2_{\mathcal{L}_1(t)}] \leq \frac{\|g\|^2_{\mathcal{L}_1(t)}}{k-1} \mathbb{E}\left[\max_{t\in T_{k-1}} Q \sum_{t\in T_{k-1}} w(t) \right] \tag{44}$$

$$\leq \frac{\|g\|^2_{\mathcal{L}_1(t)}}{k-1} \mathbb{E}\left[\max_{t\in T_{K-1}} Q \right] . \tag{45}$$

Then, we obtain the inequality in the expected excess training error,

$$\mathbb{E}[R_K] \leq \frac{2Q\|g\|^2_{\mathcal{L}_1}}{K(K-1)} \leq \frac{2M\|g\|^2_{\mathcal{L}_1}}{BK(K-1)} \simeq \frac{2M\|g\|^2_{\mathcal{L}_1}}{BK^2} . \tag{46}$$

Finally, combining Eqns. (46) and (35), we have

$$\mathbb{E}\left[\|h_{T_K,n}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2_2\right] \leq 2\|g-f\|^2_2 + 2\mathbb{E}[R_K] + C_2 \frac{2^K \operatorname{VC}(\mathcal{H})\log^2 n}{n}$$
$$\leq 2 \inf_{f\in\mathcal{F}} \left\{ \|g-f\|^2_2 + C_1 \frac{\|g\|^2_{\mathcal{L}_1}}{K^2} + C_2 \frac{2^K d\log^2 n}{n} \right\} , \tag{47}$$

for two positive constants $C_1(B, M)$ and $C_2(c_1, c_2, B, M)$. ∎

7 Experiments

In this section, we verify our theoretical results on two simulated datasets that align with our assumptions. Moreover, we evaluate `FC-ODT` on eight real-world datasets and compare it to other state-of-the-art ODTs to demonstrate its superiority.

7.1 Results on Simulated Datasets

7.1.1 IMPLEMENTATION DETAILS

Dataset　We generate two simulated datasets **sim1** and **sim2**. They are fully aligned with the assumption of our theoretical analysis, as stated in Eqn. (11). All simulated datasets are generated as $y = f(\boldsymbol{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$. For **sim1** dataset,

$$
\begin{aligned}
f(\boldsymbol{x}) = &\operatorname{ReLU}(\boldsymbol{x}_1) + \operatorname{ReLU}\left(\frac{\boldsymbol{x}_2 + \boldsymbol{x}_3}{2}\right) \\
&+ \operatorname{ReLU}\left(\frac{\boldsymbol{x}_4 + \boldsymbol{x}_5 + \boldsymbol{x}_6}{3}\right) + \operatorname{ReLU}\left(\frac{\boldsymbol{x}_7 + \boldsymbol{x}_8 + \boldsymbol{x}_9 + \boldsymbol{x}_{10}}{4}\right) \\
&+ \operatorname{ReLU}\left(\frac{\boldsymbol{x}_1 + \boldsymbol{x}_3 + \boldsymbol{x}_5 + \boldsymbol{x}_7 + \boldsymbol{x}_9}{5}\right) \ .
\end{aligned}
\tag{48}
$$

For **sim2** dataset,

$$
\begin{aligned}
f(\boldsymbol{x}) = &\exp(\boldsymbol{x}_1) + \exp\left(\frac{\boldsymbol{x}_2 + \boldsymbol{x}_3}{2}\right) \\
&+ \exp\left(\frac{\boldsymbol{x}_4 + \boldsymbol{x}_5 + \boldsymbol{x}_6}{3}\right) + \exp\left(\frac{\boldsymbol{x}_7 + \boldsymbol{x}_8 + \boldsymbol{x}_9 + \boldsymbol{x}_{10}}{4}\right) \\
&+ \exp\left(\frac{\boldsymbol{x}_1 + \boldsymbol{x}_3 + \boldsymbol{x}_5 + \boldsymbol{x}_7 + \boldsymbol{x}_9}{5}\right) \ .
\end{aligned}
\tag{49}
$$

For each simulated dataset, $\boldsymbol{x}$ is uniformly distributed over $[-3, 3]^{10}$ and $\sigma = 0.01$. Note that the test samples are generated without noise ϵ, which aligns with the definition of consistency.

Compared Methods　We choose a representative ODT method for comparison, i.e., `Ridge-ODT` (Menze et al., 2011), which uses ridge regression to learn the optimal split direction. Since `FC-ODT` also uses ridge regression for the linear projection, `Ridge-ODT` can be seen as an ablation study without feature concatenation. On the other hand, the predictive performance varying with tree depth K and number of samples n for both methods can validate our theoretical findings.

Training Details　Experiments are run on a machine with a 3.40 GHz Intel i7-13700KF CPU. To prevent overfitting risk caused by insufficient sample size in leaf nodes, the minimum number of samples to split a node is set to 20, and the minimum number of samples in leaf nodes is set to 8. The regularization parameter λ in `FC-ODT` and `Ridge-ODT` is chosen from $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ using grid search with 5-fold cross-validation on the training set.
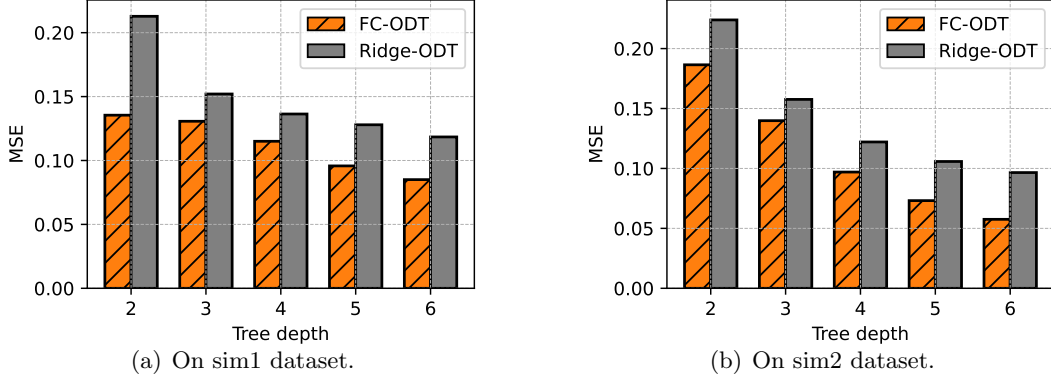
Figure 2: MSE values with different maximum tree depths.

Evaluation Protocol The performance is measured by MSE (Mean Squared Error) on the test samples. The decrease in MSE reflects convergence. Each simulated dataset is randomly generated 10 times, and the average performance is reported.

7.1.2 CONVERGENCE RATE *w.r.t.* TREE DEPTH

This experiment aims to verify the theoretical advantage of `FC-ODT` on convergence rate w.r.t. the maximum tree depth. According to Theorem 6, when n is sufficiently large, the $\mathcal{O}(1/n)$ term is sufficiently small, and the relationship between tree depth K and test error can be studied. So we set the number of training samples to 2000. Theorem 6 indicates that the test error of `FC-ODT` should be considerably better than `Ridge-ODT` with a limited tree depth, so we set tree depth $K \in \{2, 3, 4, 5, 6\}$. We compare `FC-ODT` with `Ridge-ODT` on the two simulated datasets **sim1** and **sim2**, each time 2000 training samples and 500 test samples are randomly generated. Figure 2 shows the average test MSE of `FC-ODT` and `Ridge-ODT` with the increasing tree depth. We can observe that `FC-ODT` has a faster convergence rate, which is consistent with our theoretical result.

7.1.3 CONVERGENCE RATE *w.r.t.* NUMBER OF SAMPLES

Based on Figure 2, $K = 4$ performs well. While deeper trees would result in a slight performance improvement, this comes at the cost of losing interpretability, which goes against the original intent of ODTs. Therefore, we choose $K = 4$ as the regular setting thereafter. We compare `FC-ODT` and `Ridge-ODT` with increasing number of training samples $n \in \{50, 100, 200, 500, 1000, 2000\}$ on the two simulated datasets: **sim1** and **sim2**. Figure 3 shows the test MSE of `FC-ODT` and `Ridge-ODT` with increasing number of samples. We can observe that both exhibit similar convergence tendencies as the number of training samples increases, which aligns with previous theoretical findings. Meanwhile, `FC-ODT` always has a certain advantage in performance over `Ridge-ODT`, and its superiority is more pronounced with limited training samples. It suggests that the feature concatenation mechanism has potential in handling tasks with limited samples. This practical discovery is worth theoretical analysis in future work.
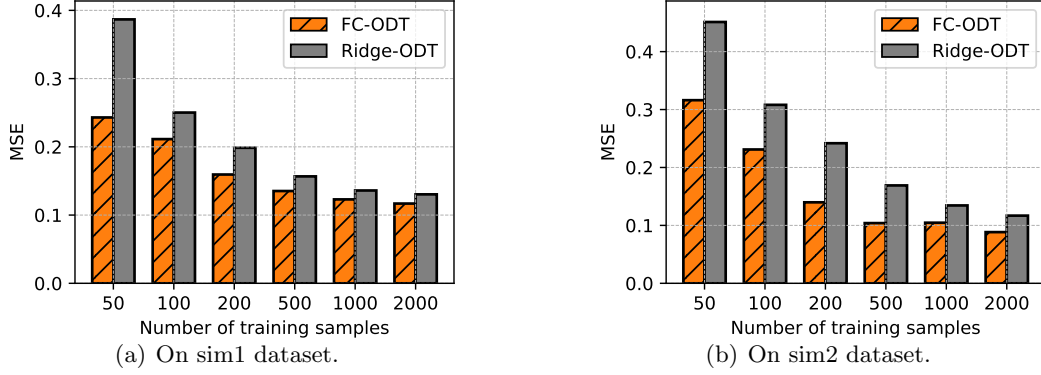
16

(a) On sim1 dataset.

(b) On sim2 dataset.

Figure 3: MSE values with different numbers of training samples.

Table 1: Dataset information.

Dataset	$N_{samples}$	$N_{features}$
sim1	2000	10
sim2	2000	10
abalone	4177	8
appliances	19735	32
auto_mpg	398	7
bodyfat	252	14
cadata	20640	8
concrete	1030	8
cpusmall	8192	12
french	105168	20
housing	506	13
pm2.5	41757	14
space_ga	3107	6
superconductivity	21263	81
mg	1385	6
mpg	392	7
msd	515345	90

7.2 Results on Real-World Datasets

7.2.1 IMPLEMENTATION DETAILS

Datasets   We conduct experiments on two simulated datasets and eight LIBSVM regression datasets (Chang and Lin, 2011). Table 1 summarizes the details of the simulated and real datasets. Each dataset is randomly split into training and test sets in a ratio of 3:2. This data split process is repeated 10 times for 10 independent runs.

17

Compared Methods   We test the R^2 of our method `FC-ODT` against `CART` and other state-of-the-art ODT methods.

- `S1O` (Zhu et al., 2020): An ODT method using mixed-integer programs (MIP) to learn optimal split directions.

- `TAO` (Carreira-Perpinán and Tavallali, 2018): An ODT method alternately updating the tree structure and linear combination to learn optimal split directions.

- `BUTIF` (Barros et al., 2014): An ODT method using embedded feature selection to learn optimal split directions.

- `Ridge-ODT` (Menze et al., 2011): An ODT method using ridge regression to learn optimal split directions.

- `EC-ODT` (Czajkowski and Kretowski, 2013): An ODT method using evolutionary computation to learn optimal split directions.

Training Details   This experiment's configurations are the same as the simulation experiment. Based on the simulation experiments, we keep the setting of tree depth $K = 4$ for all the compared methods.

Evaluation Protocol   Unlike the simulation experiments, in real-world problems, we do not know the underlying function $f(\boldsymbol{x})$. Therefore, in this experiment, we no longer compute MSE on the noise-free "test samples". Instead, we evaluate the performance on the randomly split test samples and use R^2 as the performance measure:

$$R^2 = 1 - \frac{SS_{res}}{SS_{total}} \ , \tag{50}$$

where SS_{res} is the sum of squared residuals, representing the difference between the predicted and actual values of the model, and SS_{total} is the total sum of squares, representing the difference between the actual value and the average value. The value range of R^2 is between 0 and 1. The closer R^2 is to 1, the model explains more variance and fits better.

7.2.2 R^2 SCORE COMPARISON

Table 2 reports the performance of the compared methods on seventeen datasets. The statistical significance of the experiments is tested through the Wilcoxon rank-sum test (Rosner et al., 2003). It shows that `FC-ODT` achieves the best average test R^2 score on thirteen out of seventeen datasets, and the best average rank compared to other state-of-the-art ODT methods: `Ridge-ODT`, `TAO`, `BUTIF`, `S1O` and `EC-ODT`. It is significantly better than the baseline `CART` on all seventeen datasets. As we mentioned that `Ridge-ODT` can be viewed as an ablation study without feature concatenation, the comparison between `FC-ODT` and `Ridge-ODT` shows that the feature concatenation mechanism improves performance, indicating that transmitting projection in the decision paths improves the learning efficiency. We also conducted the Friedman-Nemenyi test of the compared methods on seventeen datasets, and Figure 4 shows that the ranking of `FC-ODT` is significantly better than that of `TAO`, `BUTIF`, `S1O`, and `EC-ODT`.

Table 2: Test R^2 (avg.±std. of 10 times of running) on seventeen datasets. The best result is in bold. Considering the Wilcoxon rank-sum test with a confidence level of 0.1, we use ● (or ○) to indicate that FC-ODT is significantly better (or worse) than the corresponding method.

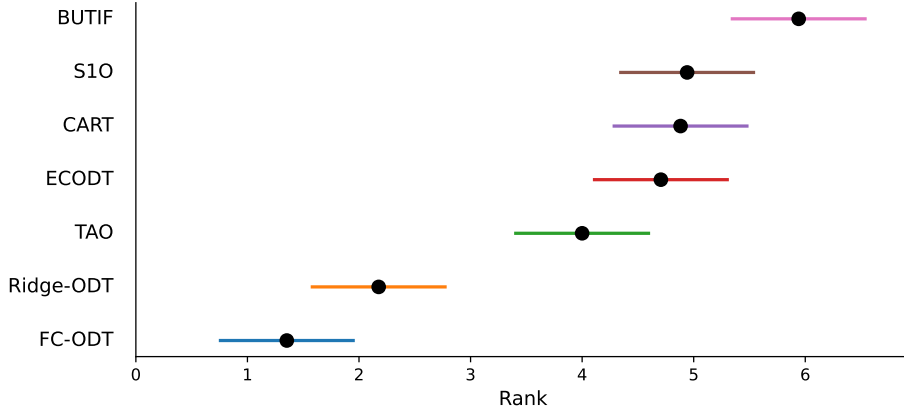| Dataset	FC-ODT	Ridge-ODT	TAO	BUTIF	S1O	CART	EC-ODT
sim1	**0.877±0.007**	0.849±0.010●	0.852±0.015●	0.772±0.013●	0.691±0.023●	0.588±0.023●	0.766±0.032●
sim2	**0.895±0.019**	0.874±0.011●	0.886±0.009●	0.799±0.010●	0.772±0.017●	0.724±0.018●	0.712±0.033●
abalone	0.541±0.021	**0.548±0.010**	0.473±0.015●	0.417±0.028●	0.467±0.020●	0.447±0.012●	0.491±0.023●
appliances	0.203±0.011	**0.207±0.016**	0.201±0.021	0.049±0.014●	0.116±0.006●	0.164±0.005●	0.162±0.010●
auto_mpg	**0.826±0.028**	0.823±0.015	0.814±0.029	0.773±0.023●	0.794±0.029●	0.803±0.021●	0.786±0.031●
bodyfat	**0.955±0.036**	0.919±0.009●	0.875±0.032●	0.818±0.036●	0.940±0.024●	0.938±0.016●	0.770±0.059●
cadata	**0.701±0.004**	0.700±0.005	0.671±0.008●	0.624±0.005●	0.552±0.012●	0.543±0.006●	0.680±0.012●
concrete	**0.732±0.038**	0.730±0.027	0.713±0.026	0.609±0.038●	0.687±0.041●	0.671±0.026●	0.707±0.028●
cpusmall	0.949±0.007	0.956±0.009○	**0.962±0.004○**	0.935±0.005●	0.939±0.002●	0.934±0.011●	0.949±0.006
housing	**0.776±0.029**	0.764±0.031	0.752±0.031●	0.701±0.047●	0.717±0.060●	0.738±0.010●	0.625±0.066●
pm2.5	**0.352±0.007**	0.344±0.006●	0.137±0.078●	0.222±0.007●	0.263±0.008●	0.261±0.009●	0.327±0.008●
space_ga	**0.583±0.035**	0.562±0.013●	0.453±0.161●	0.478±0.021●	0.501±0.029●	0.508±0.018●	0.451±0.084●
superconductivity	0.789±0.014	**0.842±0.004○**	0.798±0.009○	0.723±0.004●	0.699±0.015●	0.734±0.005●	0.763±0.010●
french	**0.943±0.001**	0.939±0.001●	0.938±0.001●	0.890±0.001●	0.877±0.004●	0.911±0.004●	0.906±0.010●
mg	**0.657±0.016**	0.640±0.018●	0.614±0.025●	0.536±0.021●	0.643±0.021●	0.625±0.021●	0.643±0.027●
mpg	**0.840±0.019**	0.833±0.021	0.833±0.020	0.789±0.036●	0.805±0.039●	0.805±0.025●	0.793±0.037●
msd	**0.306±0.001**	0.297±0.001●	timeout	0.041±0.057●	0.094±0.008●	0.142±0.001●	0.228±0.005●
average rank	1.35	2.18	4.00	5.94	4.94	4.88	4.71

Figure 4: The Friedman-Nemenyi test of the compared methods on seventeen datasets.

7.2.3 COMPUTATIONAL COMPLEXITY COMPARISON

In Figure 5, we plot the time consumption of all the compared methods. The time consumption includes hyper-parameter tuning and training process. Figure 5 show that the time consumption of FC-ODT is better than TAO and BUTIF, close to Ridge-ODT and S1O. In addition, FC-ODT outperforms EC-ODT on datasets with smaller sample sizes, but EC-ODT has less runtime on datasets with larger sample sizes. This is consistent with the results of

our analysis on time complexity in Section 4. It indicates that the feature concatenation mechanism hardly incurs any additional time overhead. However, the result shows that all the ODT methods have significantly longer running time than the axis-parallel tree, CART, mainly due to calculating the optimal linear projection. Reducing the computational complexity of optimal linear projection in ODTs is still an open problem.
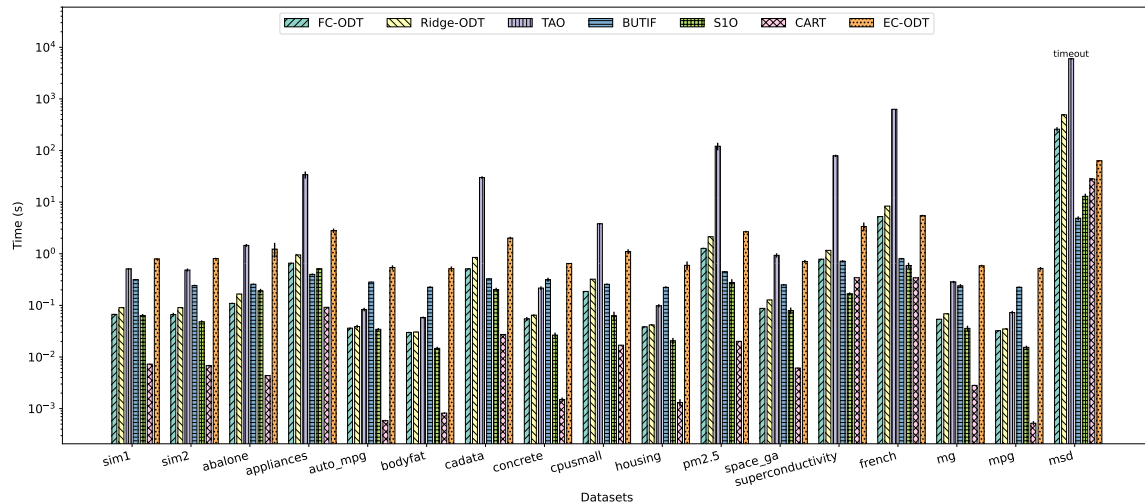

Figure 5: Running time on seventeen datasets.

8 Conclusion

This work points out that the drawback of conventional ODTs lies in the waste of projection information during the tree construction. To address it, we propose FC-ODT, which introduces a feature concatenation mechanism to transmit the projection information of the parent node through in-model feature transformation, thereby enhancing the learning efficiency. Both theory and experiments have verified that the projection information transmission brought by feature concatenation helps to improve the consistency rate. In future work, we will explore how to use random projection to enhance the diversity of concatenated features, thereby constructing an efficient random forest algorithm based on FC-ODT.

Acknowledgments and Disclosure of Funding

References

L. Arnould, C. Boyer, and E. Scornet. Analyzing the tree-layer structure of deep forests. In *Proccedings of the 37th International Conference on Machine Learning*, pages 342–350, 2021.

R. C. Barros, P. A. Jaskowiak, R. Cerri, and A. C. de Carvalho. A framework for bottom-up induction of oblique decision trees. *Neurocomputing*, 135:3–12, 2014.

P. Bartlett, Y. Freund, W. S. Lee, and R. E. Schapire. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106:1039–1082, 2017.

H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen. Decision trees: from efficient prediction to responsible ai. *Frontiers in Artificial Intelligence*, 6:1124553, 2023.

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

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

C. E. Brodley and P. E. Utgoff. Multivariate decision trees. *Machine Learning*, 19(1):45–77, 1995.

E. Cantú-Paz and C. Kamath. Inducing oblique decision trees with evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(1):54–68, 2003.

M. A. Carreira-Perpinán and P. Tavallali. Alternating optimization of decision trees, with application to learning sparse oblique trees. In *Advances in Neural Information Processing Systems 31*, pages 1211–1221, 2018.

M. D. Cattaneo, R. Chandak, and J. M. Klusowski. Convergence rates of oblique regression trees for flexible function libraries. *Annals of Statistics*, 52(2):466–490, 2024.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

M. Czajkowski and M. Kretowski. Global induction of oblique model trees: an evolutionary approach. In *Artificial Intelligence and Soft Computing*, pages 1–11, 2013.

K. Doubleday, J. Zhou, H. Zhou, and H. Fu. Risk controlled decision trees and random forests for precision medicine. *Statistics in Medicine*, 41(4):719–735, 2022.

A. Ferigo, L. L. Custode, and G. Iacca. Quality diversity evolutionary learning of decision trees. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 425–432, 2023.

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

M. A. Ganaie, M. Tanveer, P. N. Suganthan, and V. Snásel. Oblique and rotation double random forest. *Neural Networks*, 153:496–517, 2022.

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

L. Györfi, M. Kohler, A. Krzyzak, H. Walk, et al. *A Distribution-Free Theory of Nonparametric Regression*. Springer, 2002.

T. Hastie and R. Tibshirani. Generalized additive models: Some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987.

D. Heath, S. Kasif, and S. Salzberg. Induction of oblique decision trees. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1002–1007, 1993.

M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz. Comparison of arima and random forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC Bioinformatics*, 15(1):1–9, 2014.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30*, pages 3146–3154, 2017.

P. Kontschieder, S. R. Bulò, M. Pelillo, and H. Bischof. Structured labels in random forests for semantic labeling and object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2104–2116, 2014.

A. López-Chau, J. Cervantes, L. López-García, and F. G. Lamont. Fisher's decision tree. *Expert Systems with Applications*, 40(16):6283–6291, 2013.

S.-H. Lyu, L. Yang, and Z.-H. Zhou. A refined margin distribution analysis for forest representation learning. In *Advances in Neural Information Processing Systems 32*, pages 5531–5541, 2019.

S.-H. Lyu, Y.-X. He, and Z.-H. Zhou. Depth is more powerful than width with prediction concatenation in deep forests. In *Advances in Neural Information Processing Systems 35*, pages 29719–29732, 2022.

B. H. Menze, B. M. Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht. On oblique random forests. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 453–469, 2011.

S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2(1):1–32, 1994.

M. Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

N. Payet and S. Todorovic. Hough forest random field for object recognition and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1066–1079, 2012.

J. R. Quinlan. C4. 5: Programs for machine learning. In *Proceedings of 10th International Conference on Machine Learning*, pages 252—-259, 1993.

B. Rosner, R. J. Glynn, and M.-L. Ting Lee. Incorporation of clustering effects for the Wilcoxon rank sum test: A large-sample approach. *Biometrics*, 59(4):1089–1098, 2003.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

G. Smith and F. Campbell. A critique of some ridge regression methods. *Journal of the American Statistical Association*, 75(369):74–81, 1980.

T. Stepišnik and D. Kocev. Oblique predictive clustering trees. *Knowledge-Based Systems*, 227:107228, 2021.

C. J. Stone. Additive regression and other nonparametric models. *Annals of Statistics*, 13 (2):689–705, 1985.

T. M. Tomita, J. Browne, C. Shen, J. Chung, J. L. Patsolic, B. Falk, C. E. Priebe, J. Yim, R. Burns, M. Maggioni, et al. Sparse projection oblique randomer forests. *Journal of Machine Learning Research*, 21(1):4193–4231, 2020.

R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

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

Q. Wang, L. Yang, and Y. Li. Learning from weak-label data: A deep forest expedition. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 6251–6258, 2020.

L. Yang, X. Wu, Y. Jiang, and Z. Zhou. Multi-label learning with deep forest. In *Proceedings of the 24th European Conference on Artificial Intelligence*, volume 325, pages 1634–1641, 2020.

H. Zhang, A. Zhou, and H. Zhang. An evolutionary forest for regression. *IEEE Transactions on Evolutionary Computation*, 26(4):735–749, 2021.

Z.-H. Zhou and J. Feng. Deep forest: Towards an alternative to deep neural networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3553–3559, 2017.

H. Zhu, P. Murali, D. Phan, L. Nguyen, and J. Kalagnanam. A scalable MIP-based method for learning optimal multivariate decision trees. In *Advances in Neural Information Processing Systems 33*, pages 1771–1781, 2020.