REVISITING DIFFERENTIALLY PRIVATE XGBOOST: ARE RANDOM DECISION TREES REALLY BETTER THAN GREEDY ONES?

Anonymous authors
Paper under double-blind review

ABSTRACT

Boosted Decision Trees (e.g., XGBoost) are one of the strongest and most widely used machine learning models. Motivated by applications in sensitive domains, various versions of Boosted Decision Tree learners with provably differential privacy (DP) guarantees were designed. Contrary to their non-private counterparts, [Maddock et al., 2022] reported a surprising finding that private boosting with random decision trees outperforms a more faithful privatization of XGBoost that uses greedy decision trees. In this paper, we challenge this conclusion with an improved DP-XGBoost algorithm and a thorough empirical study. Our results reveal that while random selection is still slightly better in most datasets, greedy selection is not far behind after our improved DP analysis. Moreover, if we restrict the number of trees to be small (e.g., for interpretability) or if interaction terms are important for prediction, then random selection often fails catastrophically while greedy selection (our method) prevails.

1 INTRODUCTION

Gradient boosting decision trees (GBDT), proposed by [Friedman, 2001], are a well-tested tree ensemble method in data science applications. There are two types of gradient boosting decision trees based on their construction methods: (1)*Greedy Boosting*: in this approach, the tree structure is constructed by greedily minimizing a loss function. Popular implementations of this method include XGBoost [Chen & Guestrin, 2016], LightGBM [Ke et al., 2017], and CatBoost [Prokhorenkova et al., 2018]; (2) *Random Boosting*: in this method, the tree is built by randomly selecting some prefixed structure (e.g., the feature or the threshold to branch on). One well-known method in random boosting is Extra-Trees [Geurts et al., 2006].

Compared to greedy boosting, random boosting typically has lower computational costs but often requires growing more trees to achieve desirable performance. In practice, greedy boosting tends to outperform random boosting.[1]

Our paper concerns the problem of learning GBDT with Differential Privacy (DP) constraints [Dwork et al., 2006]. Surprisingly, the dynamics shift when these methods are applied in the context of differential privacy). Recent research, such as [Maddock et al., 2022], consistently demonstrates that random boosting significantly outperforms the greedy approach in the DP domain. Upon a careful review of previous DP greedy boosting trees, we found that they often suffer from either DP accounting issues or inherent flaws in mechanism design. These factors obscure the true potential of DP greedy boosting. Consequently, the question of whether DP random boosting genuinely outperforms the greedy approach remains an unresolved problem.

Summary of contributions. In this paper, we investigate the pros and cons of greedy versus random DP tree boosting. Our main contributions are twofold.

- We introduce DP-XGB, an improved DP adaptation of XGBoost, demonstrating enhanced performance through the utilization of modern DP accounting methods.

[1] A comprehensive empirical comparison between XGBoost and Extra-Trees can be found in `https://mljar.com/machine-learning/extra-trees-vs-xgboost/`

- Our large-scale evaluations reveal that DP-XGB achieves comparable performance to random boosting decision trees. (see Tab. 1) Additionally, we found two application scenarios where DP random boosting trees lag significantly behind greedy ones: classification with strong feature interactions (Fig. 2); and tasks where only a small number of trees is allowed; (Fig. 3). Our findings challenge the conventional notion that allocating privacy budgets to learning tree structures may not yield substantial benefits as emphasized in [Fletcher & Islam, 2019], [Nori et al., 2021], and [Maddock et al., 2022].

It's worth noting that, although DP random boosting with a larger number of trees yields better predictive performances, DP greedy boosting, with a much smaller number of trees, offers the interpretability of decision rules. This attribute is particularly valuable in various applications such as exploratory data analysis (including feature selection and the discovery of feature interactions), disease diagnosis [Tanner et al., 2008], and the development of fair and interpretable policies [Aghaei et al., 2019]. Consequently, we believe that DP-XGB would prove beneficial to practitioners, including medical professionals, insurers, and judges, who place a premium on explainability.

2 Motivations and Related Work

Boosting is an old idea with deep roots in computational learning theory [Schapire, 1990]. It stands out as one of the most effective machine learning algorithms, particularly when instantiated with decision trees [Friedman, 2001; Chen & Guestrin, 2016]. The literature is too vast to cover, so we refer readers to a recent survey [Sigrist, 2021] for further information.

Instead, we focus on reviewing a growing body of research on boosting decision trees with differential privacy — a well-motivated task given that XGBoost is the first choice of ML model for many researchers and data scientists who work with sensitive datasets in medical, financial, legal and public policy domains. We will review the literature on DP random boosting and DP greedy boosting, highlight potential issues for DP greedy boosting, and demonstrate recent DP accounting techniques that hold promise for enhancing performance.

DP Random Boosting, DP Greedy Boosting In the early stage of differential private decision trees, the comparison mainly revolved around DP random forest [Fletcher & Islam, 2017] and DP greedy decision trees [Friedman & Schuster, 2010]. Notably, DP random forest often demonstrated superior performance, as highlighted in a comprehensive survey by [Fletcher & Islam, 2019]. For the realm of random boosting, [Nori et al., 2021] introduced a DP explainable boosting decision tree (DP-EBM) by employing cyclical feature selection and Gaussian DP accounting [Dong et al., 2019b]. Subsequently, [Maddock et al., 2022](DP-TR) extended the DP random boosting decision tree framework and introduced several high-utility DP random boosting models, combining cyclical feature selection with an improved sketching algorithm. These advancements resulted in achieving state-of-the-art(SOTA) performance for DP Boosting trees in binary classification tasks. In the domain of DP greedy boosting, [Li et al., 2022] (DP-Boost) proposed a DP gradient boosting decision tree along with a model-averaging ensemble approach to enhance performance. [Grislain & Gonzalvez, 2021] (Sarus-XGB) integrated techniques from XGBoost, such as weighted quantile sketch and min_child_weight regularization, to create the first DP XGBoost. However, the empirical performance of DP greedy boosting has not matched that of DP random boosting. This discrepancy is attributed to the early-stage limitations of DP techniques and certain implementation issues.

Side effects that hurt DP greedy boosting (1) *Primitive privacy accounting methods:* both Sarus-XGB and DP-Boost utilize pure DP accounting for compositions, resulting an overestimation of privacy budget; (2) *Flaws in DP Mechanism Design:* Sarus-XGB and DP-Boost add noise to leaf weight through output perturbation, leading to a noisy estimator due to larger global sensitivity; (3) *Absence of Hessian information:* Neither Sarus-XGB nor DP-Boost incorporated Hessian information. they relied solely on gradient information for tree splitting and leaf weight release. This contrasts with approaches like XGBoost, which leverages Hessian information to get further performance improvements.

Improved DP accounting techniques We can solve privacy accounting issues through recently developed DP techniques. (1) *Tighter privacy accounting using Rényi-DP*: [Mironov, 2017] propose Rényi-DP, offering a mechanism-specific way to characterize privacy guarantee. Composition over the Rényi-DP domain gives tighter privacy accounting than directly using pure DP composition or

advanced composition [Dwork et al., 2010]. (2) *Improved Privacy Accounting for DP mechanisms* Recent research has also enhanced mechanism-level privacy accountings. An improved zero concentrated differential privacy (zCDP) bound for the exponential mechanism [Dong et al., 2019a], as well as better privacy accounting for the Gaussian mechanism [Balle & Wang, 2018] [Dong et al., 2019b], have been introduced. These improvements are particularly beneficial when the privacy budget is limited. Our model uses these mechanisms for tree structure and leaf release, helping us efficiently manage and save privacy budgets while maintaining strong privacy guarantees.

Thus, by mitigating side effects and harnessing improved differential privacy techniques, the question arises: Is DP random boosting really better than DP greedy boosting?

Other related work on privacy and boosting. The first differentially private boosting algorithm was introduced in [Dwork et al., 2010], which applies boosting for private query release instead of solving ML tasks, though the same algorithm theoretically works for solving private learning via the statistical query model [Kasiviswanathan et al., 2011]. Recent work of [Bun et al., 2020] designed private boosting for learning linear separately and showed that it is able to adapt to large margin. We refer to the [Bun et al., 2020] and the references therein for a more comprehensive survey of the literature on privacy and boosting from the learning theory point of view (e.g., work that extends AdaBoost [Schapire, 1990]). Finally, the recent work [Tang et al., 2023] also studied gradient boosting with differential privacy but focused on linear learners rather than decision trees.

3 PRELIMINARIES

3.1 DIFFERENTIAL PRIVACY

To begin with, we introduce some definitions from differential privacy literature.

Definition 1. *(Differential privacy [Dwork et al., 2006]) A randomized algorithm $\mathcal{M} : \mathcal{X} \to \Theta$ is (ε, δ)-DP (differential private) if for any neighbouring datasets $x, x' \in \mathcal{X}$ and a measurable set $O \subseteq \Theta$, we have $\mathbb{P}(\mathcal{M}(x) \in O) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(x') \in O) + \delta$.*

In this paper, we consider adding/removing neighboring relationships. Namely, x and x' are neighboring datasets if x is the same as x' after adding or removing a single data point.

Definition 2. *(Rényi differential privacy [Mironov, 2017]) An randomized algorithm $\mathcal{M} : \mathcal{X} \to \Theta$ is $(\alpha, \varepsilon(\alpha))$-RDP (Rényi-DP) with order $\alpha \in (1, \infty)$ if for all neighbouring dataset $x, x' \in \mathcal{X}$, we have $D_\alpha(\mathcal{M}(x)||\mathcal{M}(x')) = \frac{1}{\alpha-1}\log \mathbb{E}_{o\sim\mathcal{M}(x')}\left[\left(\frac{\mathbb{P}(\mathcal{M}(x)=o)}{\mathbb{P}(\mathcal{M}(x')=o)}\right)^{\alpha}\right] \leq \varepsilon(\alpha)$*

RDP offers a tighter and cleaner analysis for compositions. Namely, if $\mathcal{M}_1$ satisfies $(\alpha_1, \varepsilon_1)$-RDP and $\mathcal{M}_2$ satisfies $(\alpha_2, \varepsilon_2)$-RDP, the composed mechanism $\mathcal{M}_1 \circ \mathcal{M}_2$ satisfies $(\alpha_1 + \alpha_2, \varepsilon_1 + \varepsilon_2)$-RDP. RDP is also a generalization of zero-concentrated differential privacy (zCDP) [Bun & Steinke, 2016] with definition: if $D_\alpha(\mathcal{M}(x)||\mathcal{M}(x')) \leq \rho\alpha$ for any $\alpha \in (0, \infty)$, then $\mathcal{M}$ further satisfies ρ-zCDP.

Next, we introduce the exponential mechanism for learning decision tree structures:

Definition 3. *(Exponential Mechanism [McSherry & Talwar, 2007]) Let $\mathcal{H}$ be a item space and a score function $s : \mathcal{H} \to \mathbb{R}$. The exponential mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{H}$ is a randomized algorithm which outputs $h \in \mathcal{H}$ by probability $\mathbb{P}(\mathcal{M}(x) = h) = \frac{exp(-\frac{\varepsilon}{2\Delta_s}s(x,h))}{\sum_{h\in\mathcal{H}} exp(-\frac{\varepsilon}{2\Delta_s}s(x,h))}$, where the sensitivity $\Delta_s = \max_{x\sim x'\in\mathcal{X}} \max_{h\in\mathcal{H}} |s(x,h) - s(x',h)|$*

Recently, the zCDP privacy accounting of exponential mechanism has been improved by [Dong et al., 2019a] through bounded range (BR) analysis. Their methods reduce the zCDP privacy parameter from $\frac{\varepsilon^2}{2}$ to $\frac{\varepsilon^2}{8}$ for any ε-DP exponential mechanism. We extend their analysis to RDP domain and get an improved RDP bound which is smaller than the direct conversion from zCDP (Theorem 1).

Finally, we introduce the Gaussian mechanism that we use for leaf weight releasing:

Definition 4. *(Gaussian mechanism [Dwork & Roth, 2014]) The Gaussian mechanism $\mathcal{M} : \mathcal{X} \to \mathbb{R}^n$ of the form $\mathcal{M}(x) = f(x) + N(0, \Delta_q\sigma^2)$ is $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP where $\Delta_q = \max_{x\sim x'\in\mathcal{X}} \|f(x) - f(x')\|$*

3.2 GREEDY BOOSTING DECISION TREES

We give a brief introduction to the greedy gradient boosting algorithm. Consider a twice differentiable loss function $l : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and a dataset $\{x_i, y_i\}_{i=1}^n \subset \mathbb{R}^n \times \mathbb{R}$. Starting from an initial guessing F_0, gradient boosting finds a linear combination from a base learner class $\mathcal{F}$ by choosing f_{k+1} greedily at $(k+1)$-th boosting rounds and update $F_{k+1}(x)$ to be $F_k(x) + \eta f_k(x)$:

$$f_{k+1} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} l\left(y_i, F_k(x_i) + f(x_i)\right) \approx \begin{cases} \sum_{i=1}^n l\left(y_i, F_k(x_i)\right) + g_i \cdot f(x_i), & \text{(Gradient Boosting)} \\ \sum_{i=1}^n l\left(y_i, F_k(x_i)\right) + g_i \cdot f(x_i) + \frac{h_i}{2} \cdot f^2(x_i), & \text{(Newton Boosting)} \end{cases} \tag{1}$$

where $g_i = \frac{\partial l}{\partial F}|_{F=F(x_i)}$ and $h_i = \frac{\partial^2 l}{\partial F^2}|_{F=F(x_i)}$. Based on different approximation methods, greedy boosting can be classified as gradient boosting and Newton boosting. For the contents below, unless explicitly stated, we use the term "boosting" to specifically refer to Newton boosting. Notably, XGBoost represents an enhanced implementation of Gradient Boosted Decision Trees. It incorporates the Hessian-weighted quantile sketch algorithm to effectively handle high-dimensional continuous features and utilizes Hessian information for leaf release to improve predictive performance.

4 OUR GREEDY BOOSTING DECISION TREES

We present an in-depth overview of our DP-XGB design, which contains the same building blocks in the classical XGBoost algorithm: split candidate proposal, node selection, and leaf weight release.

4.1 SPLIT CANDIDATE PROPOSAL

At the beginning of each boosting round, a set of splitting thresholds is proposed by discretizing features using uniform binning, a method referred to as uniform sketching. This approach incurs zero privacy cost, given that the coordinates for each bin are data-independent.[2] However, due to the presence of skewness in feature distributions, uniform binning can introduce many uninformative split candidates, thereby diminishing the quality of uniform sketching.

To address this issue, [Maddock et al., 2022] proposes the adaptive Hessian sketch technique to refine the set of split candidates. The core concept behind this sketch algorithm is to merge bins with small Hessian weights while splitting those with large Hessian sums. This adjustment ensures that the updated split candidates contain roughly equal information. In our experiments, we observed that allocating a small privacy budget to the adaptive Hessian sketch yields substantial enhancements in prediction performance for our DP-XGB model (refer to Figure 2).

4.2 GREEDY SPLITTING NODE SELECTION

After the split candidate set is proposed, we choose the best candidate through the exponential mechanism. To evaluate the quality of the splitting threshold, we use the following score function:

$$score = \frac{(\sum_{i \in I_R} g_i)^2}{|I_R| + \lambda} + \frac{(\sum_{i \in I_L} g_i)^2}{|I_L| + \lambda} \tag{2}$$

where $I_R := \{x | x \text{ is allocated to left child branch of node } I\}$, λ is L_2 penalty weight. While in Newton boosting, the Hessian sum is used as the denominator in eqn. 2, using cardinalities of instance set instead offers an advantage of bounded global sensitivity. This advantage becomes particularly important for boosting with binary cross-entropy loss function.[3] As shown in Lemma 2 of [Li et al., 2022], this score function has global sensitivity $3\Delta_g^2$ regardless of the regularization parameter λ, which equals 3 for the cross entropy loss in our model.

4.3 LEAF RELEASING

Once the split candidates are selected, the next step is to release the leaf weight w_j^* privately. In prior approaches like Sarus-XGB and DP-Boost, this is accomplished by directly perturbing w_j^*

[2] assuming the upper and lower bounds for each feature are public information

[3] for binary cross-entropy loss, the sample Hessian is ranging from 0 to $\frac{1}{4}$

with Laplace noise. However, this method has certain drawbacks, including the looser concentration property inherited from the Laplace distribution. Additionally, $\Delta_{w_j^*}$ itself can be very large (as discussed in Appendix. B.1). Consequently, this requires the introduction of a more substantial amount of noise to ensure privacy protection.

$$\underbrace{w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}}_{\text{Non-private}}, \quad \underbrace{\hat{w}_j = w_j^* + \text{Laplace}\left(\frac{\Delta_{w_j^*}}{\epsilon}\right)}_{\text{Sarus-XGB, DP-Boost}}, \quad \underbrace{\tilde{w}_j = -\frac{\sum_{i\in I_j} g_i + \mathcal{N}(0, \tilde{\Delta}^2\sigma^2)}{\min\{\sum_{i\in I_j} h_i + \mathcal{N}(0, \tilde{\Delta}^2\sigma^2) + \lambda, \lambda\}}}_{\text{DP-TR, Ours. } (\lambda \text{ is } L_2 \text{ penalty})} \tag{3}$$

In contrast, DP-TR and our DP-XGB use the Gaussian mechanism to release a noised vector of $(\sum g_i, \sum h_i)$. This separated query release has a small sensitivity $\tilde{\Delta} = \sqrt{\Delta_g^2 + \Delta_h^2} = \sqrt{17/16}$, which offer a less noisy leaf weight. We note that this query for releasing numerator and denominator separately is also known as sufficient statistics perturbation(SSP) in DP literature [McSherry & Mironov, 2009; Vu & Slavkovic, 2009; Foulds et al., 2016; Zhang et al., 2016; Wang, 2018].

4.4 Comparison to related work

We highlight several key differences between our DP-XGB and earlier DP greedy boosting methods (DP-Boost and Sarus-XGB). We also have included a Tab. 3 in the appendix for the purpose of comparing the privatization design among different DP boosting tree models.

- (Tighter privacy guarantee) We use RDP accounting, which results in a tighter privacy guarantee, while other DP greedy boosting methods we compared (Sarus-XGB and DP-Boost) use pure DP
- (Less hyperparameter dependence) To improve performance, Sarus-XGB heavily relies on setting a large min_child_weight[4] to enrich signals within leaf nodes. However, this introduces an additional hyperparameter and incurs extra privacy costs associated with testing leaf Hessian sums and min_child_weight thresholds. Moreover, using a large min_child_weight can lead to over-regularization, occasionally harming predictive performance. Additionally, we have identified several implementation issues in Sarus-XGB, which we have detailed in Appendix B.2.
- (Use Hessian for performance enhancement) Both DP-Boost and Sarus-XGB use square loss for classification, which is essentially gradient boosting. We use cross entropy loss and achieve better performance using Newton boosting.

In contrast to DP-TR and DP-EBM, where we incorporate some shared techniques such as adaptive hessian and cyclical feature selection, our DP-XGB is methodologically different due to the use of greedy boosting.

Our DP-XGB can also be seen as a differential privacy adaptation of hybrid boosting [Friedman, 2001] since we use gradient information to guide node splitting but use both gradient and Hessian for leaf weight releasing. It's worth noting that hybrid boosting has been shown to achieve faster convergence than solely gradient-based boosting in non-private boosting literature [Sigrist, 2021].

5 Improved Differential Privacy Accounting

Privatizing tree ensembles hinges on extensive compositions of various DP mechanisms. In our DP-XGB, we use the exponential mechanism for tree node selection and the Gaussian mechanism for histogram and leaf weight release. zCDP composition offers a more privacy stringent guarantee compared to pure DP composition, enabling us to save an additional budget for the exponential mechanism, thanks to the improved zCDP bound presented in [Cesar & Rogers, 2020]. Furthermore, our finding indicates that we can achieve even greater savings on the exponential mechanism by enhancing its RDP bound also through bounded range analysis, as shown in Thm. 1:

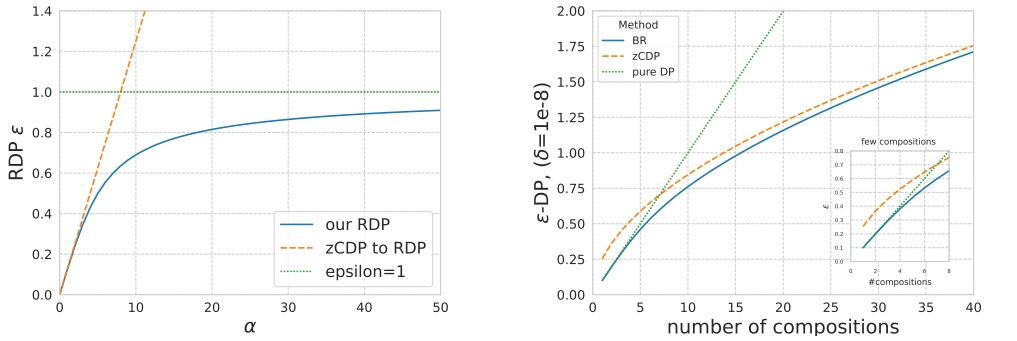
[4]see https://xgboost.readthedocs.io/en/stable/parameter.html

Theorem 1. *Let $\mathcal{M}$ be any ε bounded range mechanism. It satisfies $(\alpha, f_\varepsilon(\alpha))$-RDP with*

$$f(\alpha) = \begin{cases} \frac{\varepsilon}{e^\varepsilon - 1} - 1 - \log\left(\frac{\varepsilon}{e^\varepsilon - 1}\right), \text{ if } \alpha = 1 \\ \sup_{t \in [0,\varepsilon]} \frac{1}{\alpha - 1}\left[\alpha(t - \varepsilon) + \log((e^{\alpha\varepsilon} - 1)p_{\varepsilon,t} + 1)\right], \text{ if } \alpha > 1, \text{ where } p_{\varepsilon,t} = \frac{e^{-t} - e^{-\varepsilon}}{1 - e^{-\varepsilon}} \end{cases} \tag{4}$$

We defer proofs to Appendix A. Thm. 1 applies to the exponential mechanism, as it is also a bounded range mechanism. To make a straight comparison, we plotted our RDP bound alongside the zCDP in Dong's paper. Our bound gives a tighter characterization for the RDP curve of the exponential mechanism.

When examining the privacy guarantee of the composed exponential mechanism valued by approximate DP, we found that RDP composition yields a tighter privacy guarantee than zCDP ones (as our BR bound is always below the zCDP one). The gap is larger when the number of compositions is small. This property indicates that our RDP bound offers more privacy budget savings when small and shallow tree ensembles are used for the purpose of explainability (discussed at end of sec. 1)



(a) Improved RDP Curve

(b) Compositions of exponential mechs ($\varepsilon = 0.1$)

Figure 1: Improved RDP bound for exponential mechanism

6 Experiments

We use simulated data to assess both boosting methods in a real-world application-inspired nonlinear classification problem. Further, we perform a comprehensive comparison between two types of DP boosting methods across 18 real-world datasets. Our evaluation involves the comparison between the four best models in their paper (prefixed with TR) and the three models developed in our paper (prefixed with XGB). To distinguish differences in split candidate proposal and feature sampling methods, we use ada and cyc to indicate the use of adaptive Hessian sketch or cyclical feature selection respectively. [5] We have excluded DP Random Forest from our evaluation as it has demonstrated suboptimal performance compared to TR based Newton boosting.

Among all experiments, we maintain a constant total privacy budget of $\varepsilon = 1$, $\delta = 1/\text{number of samples}$. In the case of random boosting, we allocate an equal portion of the privacy budget to each query. However, for greedy boosting, we initially distribute the privacy budget among different building blocks and subsequently assign an equivalent amount of privacy budget to queries within the same building block. To provide more specific details, we set the allocation ratios for the adaptive Hessian sketch as *sketch: selection: leaf* $= 0.1 : 0.6 : 0.3$, while for the uniform sketch, we use *selection: leaf* $= 0.7 : 0.3$.[6]

6.1 Simulated Classification

We generated two synthetic classification problems that involve interacting features. Feature interactions of this nature are frequently encountered in real-world datasets, including but not limited to ecological data [Duncan & Kefford, 2021] and gene expression data [McKinney et al., 2006].

[5] Details of building blocks for each model can be found in Appendix C.1

[6] These budget allocation ratios were determined through preliminary experiments on the Adult dataset.

Settings. We consider two simple binary classification problems where class labels are determined either entirely or partially by feature interactions (Appendix C.2.1). Both datasets consist of three feature vectors sampled from Gaussian distributions with arbitrarily chosen means and variances. In problem 1, labels are only determined by the two-way and three-way feature interactions. In problem 2, labels are determined by both mean effects and interactions. During training time, we only provide mean effects as input to models during training since most decision tree-based methods can handle feature interactions automatically. Number of runs are 15 with 5 repetitions across 3 train/test splits.

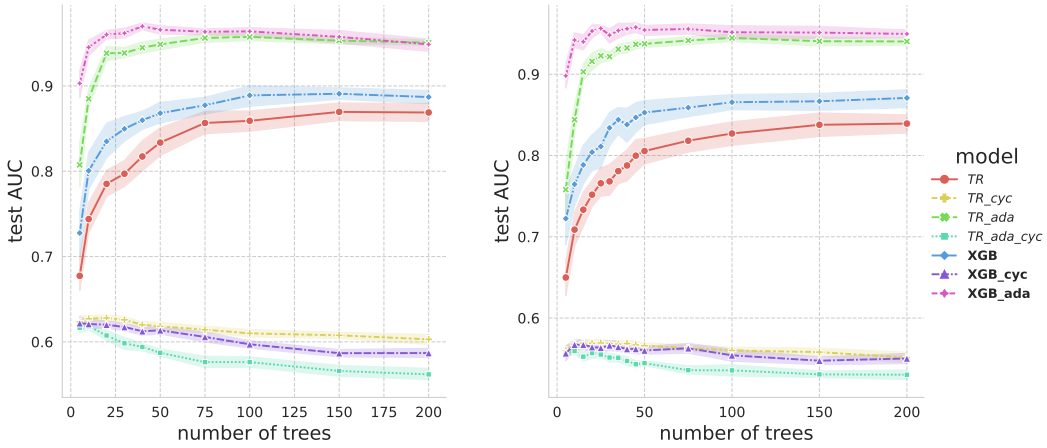


Figure 2: Simulated classification (left: problem 1, right: problem 2, shaded regions represent standard deviations.)

Results. We observe that greedy boosting consistently outperforms random boosting in both problem 1 and problem 2 (as illustrated in Figure 2). Notably, greedy boosting achieves better performance with a significantly smaller number of trees (around 35), whereas random boosting requires a much larger number of trees (typically 100 or 200) to achieve comparable results. This reinforces the importance of allocating privacy budgets to learn decision tree structures in DP Boosting.

Furthermore, we have discovered that cyclical feature sampling proves to be ineffective, irrespective of the boosting type being employed. We think this failure to be the inability of cyclical feature sampling to model feature interactions. On the other hand, our experiments show that adaptive Hessian sketch provides benefits for both greedy and random boosting.

6.2 Real Data Experimental Results

Settings. To get a thorough comparison between greedy and random boosting, we conduct experiments on a collection of 18 real-world classification datasets (Appendix C.2.2). This collection includes all seven datasets previously used in Sarus-XGB and DP-TR, as well as an additional ten numerical binary classification datasets from [Grinsztajn et al., 2022]. For hyperparameters of tree structures, we keep consistent with [Maddock et al., 2022] and set num_tree $\in \{5, 10, 15, ..., 50, 75, 100, 150, 200\}$, max_depth $\in \{2, 3, 4, 5, 6\}$. For optimization-related hyperparameters, we set the learning rate $\eta = 0.3$, L_1 penalty $\alpha = 0$, and L_2 penalty $\lambda = 1$. In order to account for the randomness in both data distributions and DP mechanisms, we split every dataset randomly into training(7): test(3) 3 times, repeat the experiment 5 times on every data slice, and report mean test AUC for each model. To investigate the change of performance influenced by the varying number of trees, we average the rank of models (ranked by mean test AUC) over all datasets (Fig. 3). The best results for each model are also included in Tab. 1

Results. When the number of trees is small (less than 50), XGB_ada outperforms or is on par with all random boosting methods (Fig. 3 left). However, for larger tree ensembles, random boosting tends to achieve better performance (Fig. 3 right). Among greedy boosting methods, XGB_ada stands out as the best model. For random boosting, the top two models are TR_cyc and TR_ada. Overall, we found DP greedy boosting is only marginally less effective than DP random boosting (Tab. 1).

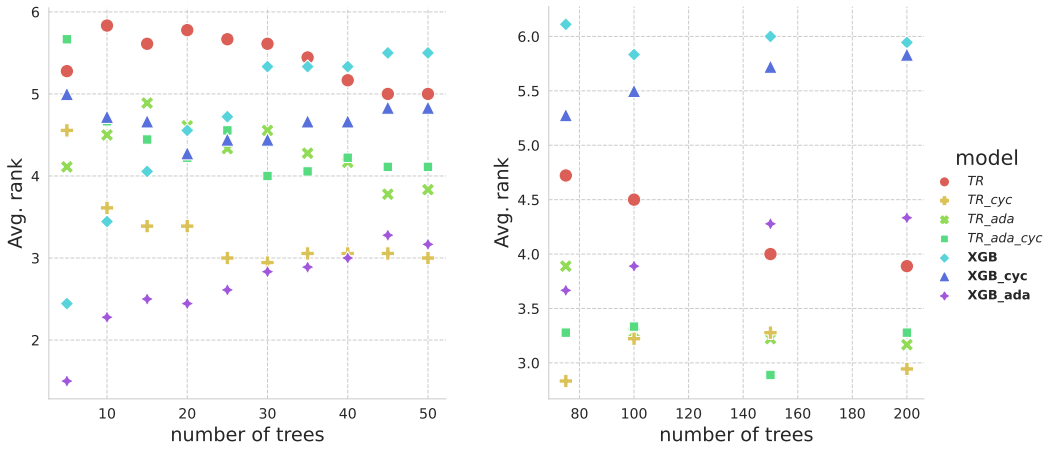


Figure 3: Model performance rank (rank 1 is best)

Table 1: Test AUC Mean and Standard Deviation for Different Models on Various Datasets

Dataset	XGB_cyc	XGB_ada	XGB	TR_cyc	TR_ada_cyc	TR_ada	TR
adult	0.8977 (0.0031)	0.8901 (0.0043)	0.8903 (0.0038)	**0.9039 (0.0019)**	0.9013 (0.0023)	0.8969 (0.0017)	0.8947 (0.0040)
aps_failure	0.9683 (0.0053)	**0.9744 (0.0043)**	0.9695 (0.0033)	0.9715 (0.0056)	0.9691 (0.0066)	0.9689 (0.0047)	0.9719 (0.0058)
bank	0.8911 (0.0043)	0.8799 (0.0049)	0.8759 (0.0049)	**0.9048 (0.0029)**	0.8994 (0.0032)	0.8853 (0.0066)	0.8896 (0.0026)
Bioresponse	0.6711 (0.0220)	0.6822 (0.0214)	0.6660 (0.0201)	**0.6906 (0.0169)**	0.6742 (0.0311)	0.6296 (0.0322)	0.6840 (0.0238)
California	0.8909 (0.0051)	0.8960 (0.0077)	0.8981 (0.0049)	0.8978 (0.0051)	0.9005 (0.0046)	0.9021 (0.0055)	**0.9030 (0.0040)**
covtype	0.8866 (0.0020)	**0.9038 (0.0050)**	0.8870 (0.0053)	0.8902 (0.0022)	0.8897 (0.0028)	0.8992 (0.0040)	0.8822 (0.0094)
credit1	0.8002 (0.0043)	0.8334 (0.0086)	0.7950 (0.0058)	0.8049 (0.0040)	0.8428 (0.0023)	**0.8435 (0.0031)**	0.8016 (0.0051)
credit2	0.7552 (0.0044)	0.7530 (0.0090)	0.7441 (0.0072)	**0.7585 (0.0039)**	0.7553 (0.0052)	0.7545 (0.0070)	0.7491 (0.0077)
Diabetes130US	0.6396 (0.0041)	0.6386 (0.0036)	0.6334 (0.0039)	0.6418 (0.0045)	0.6420 (0.0044)	**0.6410 (0.0044)**	0.6379 (0.0047)
Electricity	0.8369 (0.0037)	0.8584 (0.0035)	0.8299 (0.0050)	0.8415 (0.0039)	0.8630 (0.0019)	**0.8657 (0.0025)**	0.8377 (0.0049)
Eye_Movements	0.5655 (0.0101)	0.5578 (0.0169)	0.5589 (0.0100)	**0.5768 (0.0169)**	0.5696 (0.0138)	0.5631 (0.0129)	0.5693 (0.0157)
higgs	0.7166 (0.0021)	0.7448 (0.0031)	0.6738 (0.0110)	0.7230 (0.0018)	**0.7476 (0.0009)**	0.7423 (0.0036)	0.6758 (0.0095)
House_16H	0.9049 (0.0051)	0.9072 (0.0040)	0.8960 (0.0079)	0.9125 (0.0044)	0.9177 (0.0025)	**0.9171 (0.0050)**	0.9061 (0.0059)
Jannis	0.7902 (0.0044)	0.7997 (0.0046)	0.7770 (0.0069)	0.7978 (0.0038)	**0.8031 (0.0046)**	0.8004 (0.0047)	0.7812 (0.0069)
MagicTelescope	0.8656 (0.0060)	0.8836 (0.0045)	0.8814 (0.0066)	0.8775 (0.0026)	0.8808 (0.0036)	**0.8886 (0.0058)**	0.8856 (0.0039)
MiniBooNE	0.9250 (0.0029)	**0.9392 (0.0046)**	0.9159 (0.0071)	0.9217 (0.0063)	0.9286 (0.0031)	0.9310 (0.0039)	0.8836 (0.0278)
nomao	0.9014 (0.0027)	0.9050 (0.0019)	0.9045 (0.0022)	0.9049 (0.0026)	0.9033 (0.0031)	0.9057 (0.0022)	**0.9077 (0.0025)**
Pol	0.9410 (0.0114)	0.9533 (0.0116)	0.9620 (0.0070)	0.9436 (0.0132)	0.9418 (0.0068)	0.9473 (0.0086)	**0.9652 (0.0048)**

Note: The best result for each type of boosting has been underlined. The best result among all methods is in Boldface

7 Discussion and Future Work

When does greedy boosting outperform random ones?

- modeling feature interactions are important for prediction (Fig. 2)
- number of the trees are constrained to be small (Fig. 3 left)

Modeling feature interactions is important for classification problems, especially for data coming from domains mentioned in section 6.1. Our finding emphasizes the indispensability of DP greedy boosting in practical applications. Moreover, employing a smaller number of trees does not necessarily lead to a decrease in the performance of our DP-XGB. By examining the distribution of tree levels that fall within the best performance intervals, we found that small models (#tree$\leq$50) account for more than half of instances that reach the best performance intervals of XGB_ada and XGB_cyc (Tab. 2). This finding implies that for some of our DP greedy boosting models, achieving desirable performance can be accomplished with a limited number of trees.

An intriguing finding is that TR_cyc also performs relatively well even when the number of trees is limited. We think the reason is that splitting tree structure on a single feature yields more instances in leaf nodes, leading to signal enrichment. In addition, the extra privacy budget saved by using random boosting further reduces the magnitude of DP noise added to leaf weight. Thus, TR_cyc has more accurate leaf weights especially when using a small depth. This leads to a better performance, which aligns with the discussion in section 4.1.2 of [Nori et al., 2021]. Moreover, more than half

Table 2: Percentage% for every tree level entering best performance zone ($\varepsilon = 1$)

Method	#Tree ≤ 50	#Tree $= 75$	#Tree $= 100$	#Tree $\in \{150, 200\}$	Total
TR	13.49	3.57	5.16	13.89	36.11
TR_cyc	23.41	5.56	5.95	13.10	48.02
TR_ada	10.71	3.97	4.76	12.70	32.14
TR_ada_cyc	21.03	3.17	4.37	11.51	40.08
XGB	19.05	5.16	6.75	13.10	44.05
XGB_cyc	29.37	5.16	4.76	9.92	49.21
XGB_ada	30.56	5.16	4.76	10.71	51.19

total represents row sum, the threshold of best performance zone is $\geq$ mean_auc - std. Details in D.1

of our datasets are low-dimensional. This provides additional benefits for TR_cyc to perform well since feature selection might not be important on those datasets.

While DP random boosting can compensate for the limitation of learning feature interactions by increasing the number of trees, this approach introduces additional complexities in terms of hyper-parameter tuning. Furthermore, for deployment and explainability reasons, smaller models are often preferred. Consequently, our findings underscore the practical significance of utilizing DP greedy boosting.

What keeps DP Greedy Boosting away from better performance? We think it's the diminishing utility of the exponential mechanism when constrained by a limited privacy budget. In random boosting, the main privacy-consuming part is leaf weight release. In contrast, for greedy boosting, the privacy budget needs to be allocated to both node selection and leaf release. As the number of trees increases, the privacy budget given to the exponential mechanism decreases more rapidly than for leaf release, especially for deeper trees. This leads to the utility of the exponential mechanism resembling random selection. In such scenarios, the additional privacy budget allocated to leaf releasing plays a key role in enhancing the final performance, which explains why random boosting outperforms greedy boosting when a large number of trees are used.

Can we further improve DP-XGB? One way is to improve the utility of DP selection mechanisms, as they consume the majority of the privacy budget. We attempted to replace the exponential mechanism with permutate-and-flip [McKenna & Sheldon, 2020], which is a recently developed DP selection mechanism known for its higher utility given the same privacy budget. Unfortunately, this modification didn't yield the expected performance improvement. We believe this outcome can be attributed to the fact that the exponential mechanism can be composed through improved Rényi-DP bound (Thm. 1). However, the permutate-and-flip mechanism lacks the bounded range property due to its use of one-sided exponential noise [Ding et al., 2021]. Consequently, we can only convert from pure DP to Rényi-DP, resulting in a loose privacy accounting.

Inspired by the success of DP random boosting, it's not always necessary to greedily choose split thresholds in the building of tree structures. Instead, we can create DP mechanisms that decide whether to use a DP selection mechanism or perform random selection based on performance gains. One option is the sparse vector technique [Dwork et al., 2009], recently enhanced for Rényi-DP composition [Zhu & Wang, 2020]. Combining this with privacy filters [Feldman & Zrnic, 2021] and fully adaptive composition techniques [Whitehouse et al., 2023] enables extra budget savings by avoiding unnecessary tree structure searches. This leads to more accurate leaf weight release and potential performance improvement.

8 Conclusions

We introduce DP-XGB, an improved DP greedy boosting decision tree that leverages modern DP accounting techniques. Through a comprehensive empirical investigation, we observe that our DP-XGB is only slightly worse than the state-of-the-art DP random boosting. Additionally, in specific application scenarios such as a limited number of trees or feature interactions, the random boosting completely falls, but our method survives. Our findings challenge the conventional belief that DP random boosting surpasses DP greedy boosting and emphasize the practical irreplaceability of DP greedy boosting.

REFERENCES

Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1418–1426, 2019.

Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pp. 394–403. PMLR, 2018.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658. Springer, 2016.

Mark Bun, Marco Leandro Carmosino, and Jessica Sorrell. Efficient, noise-tolerant, and private learning via boosting. In *Conference on Learning Theory*, pp. 1031–1077. PMLR, 2020.

Mark Cesar and Ryan Rogers. Bounding, concentrating, and truncating: Unifying privacy loss composition for data analytics, 2020.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. KDD '16, pp. 785–794, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Zeyu Ding, Daniel Kifer, Thomas Steinke, Yuxin Wang, Yingtai Xiao, Danfeng Zhang, et al. The permute-and-flip mechanism is identical to report-noisy-max with exponential noise. *arXiv preprint arXiv:2105.07260*, 2021.

Jinshuo Dong, David Durfee, and Ryan Rogers. Optimal differential privacy composition for exponential mechanisms and the cost of adaptivity. *arXiv preprint arXiv:1909.13830*, 2019a.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019b.

Richard P Duncan and Ben J Kefford. Interactions in statistical models: three things to know. *Methods in Ecology and Evolution*, 12(12):2287–2297, 2021.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pp. 265–284, 2006. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pp. 381–390, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585062. doi: 10.1145/1536414.1536467. URL https://doi.org/10.1145/1536414.1536467.

Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60, 2010. doi: 10.1109/FOCS.2010.12.

Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a rényi filter. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=PBctz6_47ug.

Sam Fletcher and Md Zahidul Islam. Differentially private random decision forests using smooth sensitivity. *Expert Systems with Applications*, 78:16–31, jul 2017. doi: 10.1016/j.eswa.2017.01.034. URL https://doi.org/10.1016%2Fj.eswa.2017.01.034.

Sam Fletcher and Md Zahidul Islam. Decision tree classification with differential privacy: A survey. *ACM Computing Surveys (CSUR)*, 52(4):1–33, 2019.

James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving bayesian data analysis. *arXiv preprint arXiv:1603.07294*, 2016.

Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 493–502, 2010.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=Fp7__phQszn.

Nicolas Grislain and Joan Gonzalvez. Dp-xgboost: Private machine learning at scale. *arXiv preprint arXiv:2110.12770*, 2021.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. Privacy-preserving gradient boosting decision trees, 2022.

Samuel Maddock, Graham Cormode, Tianhao Wang, Carsten Maple, and Somesh Jha. Federated boosted decision trees with differential privacy. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2249–2263, 2022.

Ryan McKenna and Daniel Sheldon. Permute-and-flip: A new mechanism for differentially private selection. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, 2020.

Brett A McKinney, David M Reif, Marylyn D Ritchie, and Jason H Moore. Machine learning for detecting gene-gene interactions: a review. *Applied bioinformatics*, 5:77–88, 2006.

Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 627–636, 2009.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, 2007.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Harsha Nori, Rich Caruana, Zhiqi Bu, Judy Hanwen Shen, and Janardhan Kulkarni. Accuracy, interpretability, and differential privacy via explainable boosting. In *International conference on machine learning*, pp. 8227–8237. PMLR, 2021.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5:197–227, 1990.

Fabio Sigrist. Gradient and newton boosting for classification and regression. *Expert Systems with Applications*, 167:114080, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2020.114080. URL https://www.sciencedirect.com/science/article/pii/S0957417420308381.

Shuai Tang, Sergul Aydore, Michael Kearns, Saeyoung Rho, Aaron Roth, Yichen Wang, Yu-Xiang Wang, and Zhiwei Steven Wu. Improved differentially private regression via gradient boosting. *arXiv preprint arXiv:2303.03451*, 2023.

Lukas Tanner, Mark Schreiber, Jenny GH Low, Adrian Ong, Thomas Tolfvenstam, Yee Ling Lai, Lee Ching Ng, Yee Sin Leo, Le Thi Puong, Subhash G Vasudevan, et al. Decision tree algorithms predict the diagnosis and outcome of dengue fever in the early phase of illness. *PLoS neglected tropical diseases*, 2(3):e196, 2008.

Duy Vu and Aleksandra Slavkovic. Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 138–143. IEEE, 2009.

Rui Wang, Oğuzhan Ersoy, Hangyu Zhu, Yaochu Jin, and Kaitai Liang. Feverless: Fast and secure vertical federated learning based on xgboost for decentralized labels. *IEEE Transactions on Big Data*, 2022.

Yu-Xiang Wang. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. *arXiv preprint arXiv:1803.02596*, 2018.

Justin Whitehouse, Aaditya Ramdas, Ryan Rogers, and Steven Wu. Fully-adaptive composition in differential privacy. In *International Conference on Machine Learning*, pp. 36990–37007. PMLR, 2023.

Yihong Wu. Lecture 04 total variation/inequalities between f-divergences. 2016. URL http://www.stat.yale.edu/~yw562/teaching/598/lec04.pdf.

Zuhe Zhang, Benjamin Rubinstein, and Christos Dimitrakakis. On the differential privacy of bayesian inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Yuqing Zhu and Yu-Xiang Wang. Improving sparse vector technique with renyi differential privacy. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20249–20258. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e9bf14a419d77534105016f5ec122d62-Paper.pdf.

APPENDIX

A PROOF FOR RÉNYI-DP BOUNDS

We first gave a useful lemma, which bounds the Rényi divergence of the bounded range algorithm by the generalized random response (RR) (definition 2.4 in [Dong et al., 2019a]).

Lemma 1. *Let $\mathcal{M} : \mathcal{X} \to \mathbb{R}$ be any ε bounded range mechanism. For any $\alpha \geq 1$ and any neighbouring datasets x^0, x^1, the following holds:*

$$\sup_{x^0 \sim x^1 \in \mathcal{X}} D_\alpha(\mathcal{M}(x^0)||\mathcal{M}(x^1)) \leq \sup_{t \in [0,\varepsilon]} D_\alpha(RR_{\varepsilon,t}(0)||RR_{\varepsilon,t}(1))$$

Proof of Lemma 1. For fixed $x^0, x^1 \in \mathcal{X}$, by lemma 2, we have:

$$\begin{aligned} D_\alpha(\mathcal{M}(x^0)||\mathcal{M}(x^1)) &= D_\alpha(\phi \circ RR_{\varepsilon,t}(0)||\phi \circ RR_{\varepsilon,t}(1)) \\ &\leq D_\alpha(RR_{\varepsilon,t}(0)||RR_{\varepsilon,t}(1)) \quad \text{(by data-processing inequality for f-divergence, Thm 4.1 in [Wu, 2016])} \end{aligned} \tag{5}$$

By taking supreme w.r.t. x^0, x^1 and t respectively for the inequality above, we prove the result. □

Since the generalized random response is also a bounded range mechanism, the above inequality is tight when $\mathcal{M}$ belongs to the generalized random response family.

The following Lemma 2 states that the distribution of bounded range mechanism can be characterized by a simple transformation from generalized random response.

Lemma 2. *(a.k.a. lemma 4.1 in [Dong et al., 2019a])*
Let $\mathcal{M}$ be any ε bounded range mechanism. For any neighbouring datasets $x^0, x^1 \in \mathcal{X}$, there exists constant $t \in [0, \varepsilon]$ and $\phi : \{0, 1\} \to \mathcal{Y}$, which both depend on $\mathcal{M}, x^0, x^1$, such that for any $y \in \mathcal{Y}$ and $b \in \{0, 1\}$, we have the following holds:

$$\mathbb{P}[\mathcal{M}(x^b) = y] = \mathbb{P}[\phi \circ RR_{\varepsilon,t}(b) = y]$$

Based on the two lemma above, we prove the theorem 1 now:

***Proof of theorem** 1.* The proof is based on bounding the RHS in Lemma 2. For $\alpha = 1$, the bound is by directly using the maxkl(ε) bound from [Dong et al., 2019a], page 34. For $\alpha > 1$, by direct calculation:

$$\begin{aligned} D_\alpha(RR_{\varepsilon,t}(0)||RR_{\varepsilon,t}(1)) &= \frac{1}{\alpha - 1} \log \left(q_{\varepsilon,t}^\alpha p_{\varepsilon,t}^{1-\alpha} + (1 - q_{\varepsilon,t})^\alpha (1 - p_{\varepsilon,t})^{1-\alpha} \right) \\ &= \frac{1}{\alpha - 1} \log \left(e^{\alpha t} p_{\varepsilon,t} + e^{\alpha(t-\varepsilon)} (1 - p_{\varepsilon,t}) \right) \\ &= \frac{1}{\alpha - 1} \left[\alpha(t - \varepsilon) + \log((e^{\alpha\varepsilon} - 1) p_{\varepsilon,t} + 1) \right] \end{aligned} \tag{6}$$

the second line above uses the fact that $e^{-t} q_{\varepsilon,t} = p_{\varepsilon,t}$ and $e^{t-\varepsilon}(1 - p_{\varepsilon,t}) = 1 - q_{\varepsilon,t}$ □

A.1 Numerical Stable implementation of Theorem 1

To implement Theorem 1, the tricky part is to calculate $\log((e^{\alpha\varepsilon} - 1)p_{\varepsilon,t} + 1)$ since $\alpha\varepsilon$ can be large when choosing large value of α. Notice that:

$$\begin{aligned} \log((e^{\alpha\varepsilon} - 1)p_{\varepsilon,t} + 1) &= \log \left((e^{\alpha\varepsilon} - 1) \frac{e^{\varepsilon - t} - 1}{e^\varepsilon - 1} + 1 \right) \\ &= \log \left(\frac{1}{e^\varepsilon - 1} \left(e^\varepsilon - 1 + (e^{\alpha\varepsilon} - 1)(e^{\varepsilon - t} - 1) \right) \right) \\ &= \log(e^{\alpha\varepsilon + \varepsilon - t} + e^\varepsilon - e^{\varepsilon - t} - e^{\alpha\varepsilon}) - \log(e^\varepsilon - 1) \end{aligned} \tag{7}$$

Thus, we can apply the log-sum-exp trick to the first term (subtract $e^{\alpha\varepsilon}$) and avoid the overflow issue.

B More detailed review of related work

B.1 Sensitivity of Leaf weight for Newton boosting with cross-entropy loss

For binary labels $y_i \in \{0, 1\}$, the classification problem is as follow:

$$\begin{aligned} \log \left(\frac{\mathbb{P}(y = 1|x_i)}{\mathbb{P}(y = 0|x_i)} \right) &= F(x_i) \\ \Rightarrow \log \left(\frac{\mathbb{P}(y = 1|x_i)}{1 - \mathbb{P}(y = 1|x_i)} \right) &= F(x_i) \\ \Rightarrow \hat{y}_i := \mathbb{P}(y = 1|x_i) &= \frac{e^{F(x_i)}}{1 + e^{F(x_i)}} \end{aligned} \tag{8}$$

where F is the output of the XGBoost model. Then, binary classification loss gives:

$$\begin{aligned} l(y_i, \hat{y}_i) &= -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i) \\ &= -y_i F(x_i) + \log\left(1 + e^{F(x_i)}\right) \\ \frac{\partial l}{\partial F} &= -y + \frac{e^F}{1 + e^F} \in [-1, 1] \\ \frac{\partial^2 l}{\partial F^2} &= \frac{e^F}{1 + e^F} \cdot \frac{1}{1 + e^F} \in \left[0, \frac{1}{4}\right] \end{aligned} \tag{9}$$

Consider a leaf node with only one sample, the leaf weight is $\frac{\partial l}{\partial F} / \left(\frac{\partial^2 l}{\partial F^2} + \lambda\right)$. When the true label $y = 1$, and let $F \to -\infty$, we have $\frac{\partial l}{\partial F} / \left(\frac{\partial^2 l}{\partial F^2} + \lambda\right) \to -1/\lambda$, which is large if using a small L_2 regularization weight λ or even unbounded if does not use L_2 regulzaization at all.

B.2 Issues in [Grislain & Gonzalvez, 2021]

We found two issues in DP-XGBoost from [Grislain & Gonzalvez, 2021]. The first one is on the implementation of the exponential mechanism. They operate the exponential mechanism on a candidate set consisting of the best split candidates of every feature gathered non-privately.[7] In addition, their method relies on tuning minimal Hessian sum for leaf node splitting to achieve good performance. But they directly compare the Hessian sum with the minimal Hessian threshold(i.e. min_child_weight), which is not private.[8]

B.3 More on related work

B.3 More on related work

For random selection, DP-TR [Maddock et al., 2022] is the current state-of-the-art among all DP random boosting and DP greedy boosting methods. Thus, our task was reduced to compare the performance between our method DP-XGB and the best methods proposed in [Maddock et al., 2022].

FEVERLESS [Wang et al., 2022] is another related DP greedy boosting decision tree model. It does not have the three issues mentioned in section 2. FEVERLESS uses Argmax post-processing on privately released feature histograms to select tree structures. This method has limited utilities because the release of too much unnecessary information introduces more noise in node selection and leaf release.

Table 3: Comparison of privatization among different models

Models	split candidate proposal	node selection	leaf releasing	DP accounting
SARUS [Grislain & Gonzalvez, 2021]	Laplace noisy histogram	exponential mechanism	Output perturbation by Laplace noise	Pure DP
DPBoost [Li et al., 2022]	Not reported in paper	exponential mechanism	Output perturbation by Laplace noise	Pure DP
FEVERLESS [Wang et al., 2022]	Gaussian noisy histogram	post-processing of histograms	post-processing of histograms	Local DP
DP-EBM [Nori et al., 2021]	Gaussian noisy histogram	random selection	Numerator: Gaussian mechanism Denominator: post-processing of histogram	Gaussian DP
DP-TR [Maddock et al., 2022]	Data independent grid or Gaussian noisy histogram	random selection	Separate Gaussian release*	RDP
Ours	Data independent grid or Gaussian noisy histogram	exponential mechanism	Separate Gaussian release*	RDP

*:i.e. release numerator and denominator separately by Gaussian mechansim

[7] see line 1273-1281 in /src/tree/updater_histmaker.cc

[8] see line 1465 in /src/tree/updater_histmaker.cc

C EXPERIMENT DETAILS

C.1 MODEL INCLUDED IN OUR EXPERIMENT

We didn't include XGB_ada_cyc since its performance is similar to XGB_ada or XGB_cyc.

Table 4: Models used in experiment

Method	Node Selection	feature selection	split candidate proposal	leaf release
TR	random	-	uniform	newton (SSP)
TR_cyc	random	cyclical	uniform	newton (SSP)
TR_ada	random	-	adaptive Hessian	newton (SSP)
TR_ada_cyc	random	cyclical	adaptive Hessian	newton (SSP)
XGB	exponential mechanism	-	uniform	newton (SSP)
XGB_cyc	exponential mechanism	cyclical	uniform	newton (SSP)
XGB_ada	exponential mechanism	-	adaptive Hessian	newton (SSP)

Note: italic font refer to methods developed in [Maddock et al., 2022], boldface texts are methods proposed in this paper

C.2 DATA

C.2.1 SIMULATED DATA GENERATION

We let $X_1 \sim \mathcal{N}(1, 25)$, $X_2 \sim \mathcal{N}(-5, 8)$, $X_3 \sim \mathcal{N}(-2, 7)$ and generate 10000 samples from X_1, X_2, X_3 respectively. Labels are generated by the indicator function $\mathbb{I}\{\frac{\exp(F(\mathbf{x}))}{\exp(F(\mathbf{x}))+1} > 0.5\}$, where $F(\cdot)$ is defined as follow:

$$\text{(Problem 1)} \quad F(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3$$
$$\text{(Problem 2)} \quad F(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_1 + x_2 + x_3$$

In addition, We project $F(\cdot)$ into $[-12, 12]$ to avoid numerical overflow issues.

C.2.2 DESCRIPTION OF REAL DATASET

Table 5: Dataset

Name	#samples	#features	% positive cases	tested in
Adult	21113	14	24.9	[Maddock et al., 2022]
Bank	31647	16	11.7	[Maddock et al., 2022]
Covtype	70000	54	66.8	[Grislain & Gonzalvez, 2021]
Credit1	84188	10	6.95	[Maddock et al., 2022]
Credit2	21000	23	22.1	[Maddock et al., 2022]
Higgs(subsampled)	140000	28	52.8	[Maddock et al., 2022]
Nomao	24125	6	28.6	[Maddock et al., 2022]
Aps Failure	22,800	170	1.81	-
Electricity	38,474	7	0.5	[Grinsztajn et al., 2022]
Eye Movements	7,608	20	0.5	[Grinsztajn et al., 2022]
California	20,634	8	0.5	[Grinsztajn et al., 2022]
MagicTelescope	13,376	10	0.5	[Grinsztajn et al., 2022]
Diabetes130US	71,090	7	0.5	[Grinsztajn et al., 2022]
Bioresponse	3,434	419	0.5	[Grinsztajn et al., 2022]
Jannis	57,580	54	0.5	[Grinsztajn et al., 2022]
MiniBooNE	72998	50	0.5	[Grinsztajn et al., 2022]
Pol	10082	26	0.5	[Grinsztajn et al., 2022]
House_16H	13488	16	0.5	[Grinsztajn et al., 2022]

D EXPERIMENT RESULT

D.1 CALCULATION METHODS FOR TABLE 2

The purpose of Tab. 2 is to check whether using a small number of trees (≤ 50) can achieve "nearly the best performance" for every model. We set the criteria of being in the best performance zone as having mean_test_AUC larger than mean_test_AUC* - std*. For each dataset, given a type of model, mean_test_AUC* (and corresponding std*) is selected from the hyperparameter configuration that achieves the highest mean test AUC. After that, we calculate the frequency of $\#trees$ from models that belong to the best performance zones across all datasets.

D.2 OVERALL PERFORMANCE

We present an overall comparison in Fig. 4. In each subplot, every data point indicates the highest mean test AUC score achieved by a specific model with a fixed number of trees (i.e. taking maximum over depths).

Figure 4: Results for number of tree ≥ 20 ($\varepsilon = 1$)

D.3 COMPARISON BETWEEN RANDOM SELECTION AND EXPONENTIAL MECHANISM WHEN NUMBER OF INFORMATIVE FEATURE IS LIMITED

(We add this section to further address questions from reviewer jsJy and 3J8a)

In this experiment, we established different configurations by varying the number of trees as $\{5, 10, 50, 100\}$ and the maximum depth as $\{1, 2, 5\}$. The data were generated with a specific focus on having a limited number of informative features, while the rest were created using Gaussian noise. Specifically, we first generated the informative features and associated classification labels using the $sklearn.datasets.make\_classification$ function. Then, all remaining features were generated using Gaussian noise. It's important to note that we kept the total number of features constant at 100, while we explored varying the number of informative features within the set $\{2, 10, 30\}$.

In Figure 5, we observe that greedy selection consistently delivers strong performance with a small number of trees in all settings. On the other hand, for random selection, success depends on encountering informative splits. Consequently, increasing both the number of guesses (calculated as the product of the number of trees and maximum depth) and the number of informative features would increase the chance of achieving this success.

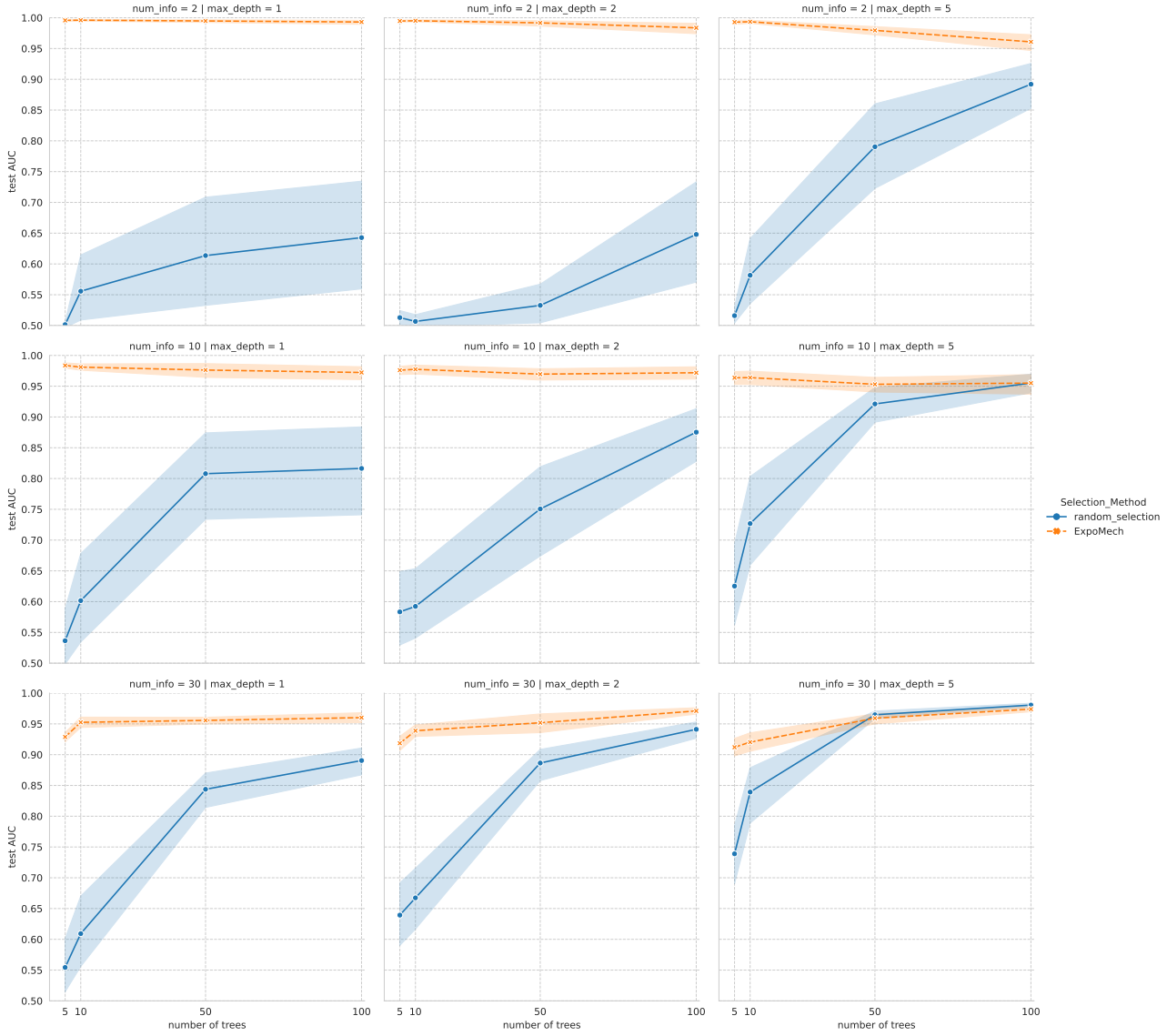


Figure 5: Comparison between exponential mechanism and random selection ($\varepsilon = 1$)

D.4 Comparison to Corollary 3.1 and Theorem 5 in [Dong et al. (2019a)]

(We add this section to further address questions from reviewer WvUe)

Our RDP analysis of bounded range mechanism (BR) gives substantial improvement over both Cor. 3.1 and Thm. 5 in [Dong et al., 2019a]. (In both graphs below, we set the base mechanism to be 0.1-DP, and in the right graph below, we set number of composition to be 20.)

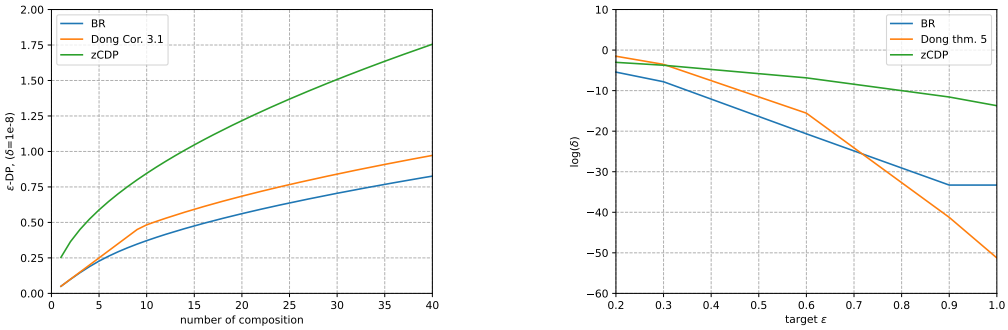


Figure 6: Comparison to Dong's corollary 3.1 and thm 5 (Updated)

D.5 Description of our algorithm

(We add this section to describe our algorithm as requested by reviewer QR2x)

Algo. 1 describe the general workflow of our method. Algo. 3 describes how we create a single decision tree.

Algorithm 1: DP Greedy Boosting

Input: Dataset D, number of trees T, tree depth d, number of split candidates Q per feature, number of feature p, split proposal for every feature $\{S_k\}_{k\in[p]}$, privacy budget ε, δ, privacy budget ratio r_1, r_2, r_3 (histogram, exponential mechanism, leaf),

Start:

$\sigma_{hist}, \varepsilon_{expo}, \sigma_{leaf} \leftarrow$ *CalibrateBudget*$(T, d, Q, p, r_1, r_2, r_3)$; `/* By AutoDP */`

$F \leftarrow F_0$

for $t = 1, \ldots, T$ **do**

  Calculate $\{g_i, h_i\}_{i\in D}$ based on F_{t-1} and store them in GH

  $\tilde{H} \leftarrow$ *GetNoisyHist*$(\{S_k\}_{k\in[p]}, GH, \sigma_{hist})$

  $F \leftarrow F + \eta \cdot BuildSingleTree(GH, \tilde{H}, \sigma_{hist}, \varepsilon_{expo}, \sigma_{leaf})$

end

Return Tree ensemble F

We set each S_k to be a uniform grid between maximum and minimum feature values[9]. Thus, the split proposal is data independent and has no privacy leakage We use the Gaussian mechanism to release noised gradient and hessian information aggregated by the split proposal.

[9] Assume upper and lower bounds for each feature are public information

Algorithm 2: GetNoisyHist

Input: split proposal for every feature $\{S_k\}_{k\in[p]}$, Raw gradient and hessian information GH, std for Gaussian mechanism on leaf histogram σ_{hist}

Start:

$\tilde{H} \leftarrow \phi$

for $k = 1, ..., p$ **do**
- $H_k \leftarrow$ hessian histogram for GH using feature k with grid S_k
- $\tilde{H}_k \leftarrow H_k + \mathcal{N}(0, \Delta^2_{hessian}\sigma^2_{hist}\mathbb{I}_Q)$
- $G_k \leftarrow$ gradient histogram for GH using feature k with grid S_k
- $\tilde{G}_k \leftarrow G_k + \mathcal{N}(0, \Delta^2_{gradient}\sigma^2_{hist}\mathbb{I}_Q)$
- $\tilde{H} \leftarrow \tilde{H} \cup \{\tilde{H}_k, \tilde{G}_k\}$

end

Return $\tilde{H}$

Algorithm 3: BuildSingleTree

Input: Raw gradient and hessian information GH, Noised histogram $\tilde{H}$, L_2 penalty weight λ, maximum depth d

for *e = 0 , . . . , d* **do**
- **for** $q \in \{$*nodes at depth e*$\}$ **do**
  - $I_q^{(e)} = \{x : x \in \{$node p in depth e$\}\}$
  - **if** *q == max_depth* **then**
    - $(\tilde{h}_q, \tilde{g}_q) \leftarrow \left(\sum_{i\in I_q^{(e)}} h_i, \sum_{i\in I_q^{(e)}} g_i\right) + \mathcal{N}(0, \sigma^2_{leaf}\mathbb{I}_2)$
    - leaf_weight$_q \leftarrow -\frac{\tilde{g}_q}{max\{\tilde{h}_q+\lambda,\lambda\}}$
    - Add leaf node to the current tree
  - **else**
    - Operate Exponential Mechanism on $\tilde{H}$ to select feature and threshold to split: $(\widehat{fea}, \widehat{thr})$; `/* use Eqn 2 as score function */`
    - Left_child$_q$ =$\{$x $\in I_q^{(e)} : x[\widehat{fea}] \leq \widehat{thr}\}$
    - Right_child$_q$ =$\{$x $\in I_q^{(e)} : x[\widehat{fea}] > \widehat{thr}\}$
    - Add children nodes to the current tree
  - **end**
- **end**

end

Return Tree f