IDENTIFYING INTERACTIONS AMONG CATEGORICAL PREDICTORS WITH MONTE-CARLO TREE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Identifying interpretable interactions among categorical predictors for predictive modeling is crucial in various research fields. Recent studies have examined interpretable interactions using decision tree (DT) learning methods, which construct DTs by greedy rules due to the high memory and time complexity of building and evaluating DTs, resulting in a local optimal solution. This paper formulates the selection of quadratic and higher order interactive terms into a LASSO problem and then relaxes it into multiple DT learning problems. A Monte Carlo Tree Search-based interaction selection (MCTs-IS) method is proposed to identify the optimal DT in an online learning manner. A DT pruning strategy is developed based on LASSO that can easily be applied to MCTs. We prove that MCTs-IS converges with high probability to the optimal solution of the DT learning problem. Extensive experiments have been conducted to demonstrate the effectiveness of the proposed algorithm on real-world datasets.

1 INTRODUCTION

n-way interaction among categorical predictors occurs when values of n categorical predictors depend on each other, or n categorical predictors affect the variability of the response variables in a nonlinear form (Lian et al., 2018). Understanding interactions provides domain knowledge about how the predictors function together to vary the target. Nevertheless, even a small set of categorical predictors can generate a large number of candidate interactions because of the combinatorial explosion, i.e., n binary features have 2^n possible n-way interactions. For high-dimensional features, considering high-way interactions dramatically increases the number of trainable parameters in a predictive model, causing the curse of dimensionality.

Nonlinearity models such as Factorization Machines (FM) (Rendle, 2010; Sun et al., 2021; Pande, 2020) and Deep Neural Networks (DNNs) (Cheng et al., 2016; Qu et al., 2016; Guo et al., 2017; Lian et al., 2018; Tao et al., 2020; Liu et al., 2020) have been applied to map predictors and/or weights to a low-dimensional space to avoid the curse of dimensionality, which can also be used to compute interactions and predict response variables. The nonlinear dimensionality reduction, however, used in these approaches does not ensure convergence, and as a consequence, does not generate stable low-dimensional representations and cannot identify critical interactions from candidates. These are critical considerations for research areas such as ad-click recommendation (Tsang et al., 2020) and AI gaming techniques (Silver et al., 2016).

Interactive terms can be automatically selected by taking candidate terms as input features and imposing a regularizer to penalize models that use many interactive terms. The regularizer helps eliminate redundant interactions from the models (Bien et al., 2013; Lim & Hastie, 2015; Vaughan et al., 2020). However, these methods can only exclude terms from a pre-given set of interactive terms rather than discover new interactions. Because DNNs are capable of extracting critical features from raw data, black-box interaction detection algorithms (Tsang et al., 2017; 2020) identify interactive terms based on the trained network parameters with heuristic rules. The high nonlinearity of DNNs, however, makes it difficult to interpret the identified interactions even though they are explicitly modeled. Besides the lack of interpretability, random exploration is necessary to find a better local minimizer for the nonconvex loss function, which then leads to unstable output.

The tree-based interaction selection methods can effectively model categorical variables and their interactions with decision trees (DTs), and have received increasing attention in recent years.

(Sorokina et al., 2008; He et al., 2014; Luo et al., 2019; Bao et al., 2020). Instead of excluding interactions from pre-given ones, tree based methods explore new interactions by identifying higher-order interactions based on lower-order ones. Their convergence guarantees ensure stable solutions. Furthermore, features' candidate values appearing together in a traversal path of DT are interacting with one another, since the condition of a child node is predicated on the condition of the parent node. Therefore, interaction identified by DTs can be easily interpreted (He et al., 2014; Bao et al., 2020). These methods are NP-complete to learn the optimal DT (Laurent & Rivest, 1976). Therefore, DT construction algorithms rely on heuristics such as the greedy algorithm and are not guaranteed to yield the global optimal DT (Kretowski & Grzes, 2005).

Contributions. We propose a Monte-carlo tree search based interaction selection (MCTs-IS) method to select interactions among categorical predictors. Our main contributions are summarized as follows.

We formulate the high-way interaction selection problem with the safe-screen Least Absolute Shrinkage and Selection Operator (safe-screen LASSO) (Xiang et al., 2016), which identifies a subset of features receiving zero weights in the vanilla LASSO solution using significantly less computing and memory resources. The safe-screen LASSO problem is then relaxed into several DT learning problems, and a DT pruning strategy is derived from the screening criteria for the safe-screen LASSO. As pruning strategies from other methods are usually derived from local optimal DT construction methods, interactions that occur in the global optimal solution may be discarded. By contrast, our approach eliminates interactions within pruned DT's nodes in the optimal solution of the vanilla LASSO, which does not produce a local optimal solution.

We propose MCTs-IS to learn the optimal DT in each DT learning problem. In MCTs-IS, Monte Carlo Tree search (MCTs) (Kocsis et al., 2006), a framework approximating the optimal choices in exponentially large search spaces, is applied to search the optimal DT iteratively in a space- and computation- efficient manner. Specifically, MCTs-IS is trained iteratively on data instances presented one-by-one in an *online feature selection* style, and in each iteration, only a subset of features in the data instance is used to update the model parameters. In this way, MCTS-IS can update model by accessing and saving a limited number of features. Finally, we give the proof showing that with enough iterations, MCTs-IS converges to the optimal DT with high probability.

Extensive experiments are performed on datasets with large samples (from $\sim 1.9M$ to $\sim 40M$) to confirm the convergence of selected interactions of MCTs-IS. We then show that benefiting from the safe-screen LASSO and MCTs, MCTs-IS can be efficiently trained, and the performance of predictive models can be improved with the help of the selected interactions.

Related Work. To deal with the large number of possible interactions, penalization methods, such as Hierarchical LASSO (Bien et al., 2013; Vaughan et al., 2020) and Group-LASSO (Lim & Hastie, 2015), have been proposed for selecting interactions based on hierarchy assumptions, which remove high-order interactions when the related low-order interactions have been eliminated. FM-based methods differ from penalization methods in that they fit all candidate interactions with fewer parameters (Rendle, 2010; Pande, 2020; Sun et al., 2021). In practice, however, these methods usually consider two or three-way interactions due to the complexity of identifying high-way interactions.

DNNs have shown great success as powerful tools for obtaining feature representations in learning high-way interactions. Interaction selection methods such as Wide&Deep (Cheng et al., 2016), Product-Based Neural Networks (PNN) (Qu et al., 2016), Factorization-Machine based neural network (DeepFM) (Guo et al., 2017) and AutoDeepFM(Liu et al., 2020) fit interactions in an implicit fashion, where the formats of the learned interactions are not clear. As a way of learning explicit DNN interactions, eXtreme Deep Factorization Machine (xDeepFM) (Lian et al., 2018) models the interactions based on DNNs with specifically designed architectures. Rather than identify specific interactions, Neural Interaction Detection (NID) (Tsang et al., 2017) and Global Interaction Detection and Encoding for Recommendation (GLIDER) (Tsang et al., 2020) detect predictors interacting with each other by checking for non-additive effects in the non-linear activation functions of DNNs.

In addition to DNNs, tree-based methods are also effective in identifying high-way interactions. They model interactions using DTs, which are then selected according to various tree construction methods including gradient boosting decision tree (GBDT) (He et al., 2014), classification and regression tree (CART) (Sorokina et al., 2008), and the successive mini-batch gradient descent (SM-

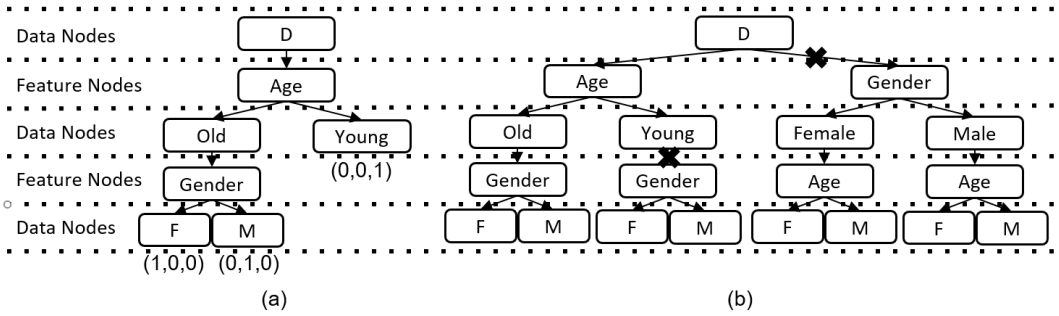

Figure 1: Examples of (a) DT and (b) FDT.

BGD) (Luo et al., 2019). Recently, the tree-based method has been successfully fused with DNNs in ResFusion to improve the performance(Bao et al., 2020).

Organization. The remainder of this paper is structured as follows. In Section 2, the interaction selection problem is formulated with LASSO and then relaxed to several DT construction problems. MCTs-IS is described in Section 3 to search the optimal DT, followed by a theoretical analysis section, where the convergence of MCTs-IS is discussed. In Section 5 we present experimental results and conclude the paper with a brief discussion of the future work in the last section.

2 The Problem Formulation

In this section, we first show how to relax the LASSO-based interaction selection problem into DT learning problems. Then we derive the divide-and-conquer approach to obtain the optimal DT in each DT learning problem. All proofs of theorems in this section are provided in Appendix A.

2.1 Selecting interactions with DTs

We assume that each (x, y) in dataset $\mathcal{D}$ is sampled from an unknown but fixed distribution, where $y \in \mathbb{R}^+$ is the response value, and $x = [x_1, x_2, ..., x_q]$ is a vector of categorical predictors. Without loss of generality, we denote the candidate values of the i^{th} predictor of x are integers ranging from 1 to K_i. Before introducing the LASSO problem, we first describe how DTs model interactions.

Definition 1 (DT)**.** *Fig. 1a describes a DT, whose nodes in odd depths are called **data nodes**, while those in even depths are called **feature nodes**. Each data node n^d stores a dataset S_{n^d}. The root node is a data node, and the depth of the root node n_r is 1, thus $S_{n_r} = \mathcal{D}$. Each data node n^d has **at most one** feature node n_i^f, whose index $i \in \{1, 2, ..., q\}$ is unique among all feature nodes along the path from DT's root node to n^d. Each feature node divides its parent node's dataset according to the i^{th} predictor of x, and then saves these sets in its K_i child data nodes. The j^{th} child node $n_{i,j}^d$ contains data $S_{n_{i,j}^d}$ whose i^{th} categorical predictor equals j. In the following section, we omit writing the subscript of $n_{i,j}^d$ and n_i^f if no confusion can arise.*

For a DT T, the leaf nodes are indexed by 1 to l_T, where l_T is the total number of the leaf nodes in T. According to Definition 1, each $(x, y) \in \mathcal{D}$ can be assigned to one and only one leaf node, whose index is marked as l_x^T. In DTs, predictors that appear in a traversal path interact with one another. Let us take the DT in Fig. 1a as an example. The leaf node marked (1,0,0) stands for the interaction between *age=old* and *gender=female*. Since each leaf node represents an interaction among predictors, T can be regarded as a model to traverse specific predictor interactions. An interaction can be expressed by a one-hot encoding vector, consisting of 0s in all l_T cells except for a single leaf node in the l_x^T cell.

Assume there are totally N_T trees that are used to transfer a data point x into N_T one-hot encoding vectors, i.e., $x^{T_1}, x^{T_2}, ..., x^{T_{N_T}}$. To exam the performance of the transformed interactions, we first model the relationship between labels and interactions with a linear model $y = \sum_{i=1}^{N_T} \sum_{j=1}^{l_{T_i}} w_j^{T_i} x_j^{T_i} + \epsilon_{(x,y)}$, where $x_j^{T_i} \in \{0, 1\}$ is the j^{th} element of x^{T_i}, $w_j^{T_i}$ is the

trainable weights of $x_j^{T_i}$, and $\epsilon_{(x,y)}$ is the error variable adding noise to the linear relationship. LASSO is then used to fit the model to select meaningful interactions, since weights of unimportant interactions will be set as zero, and thus be discarded in LASSO. Rather than minimizing $\mathbb{E}_{(x,y)\sim\mathcal{D}}(y - \sum_{i=1}^{N_T}\sum_{j=1}^{l_{T_i}} w_j^{T_i} x_j^{T_i})^2 + \sum_{i=1}^{N_T}\sum_{j=1}^{l_{T_i}} |w_j^{T_i}|$ directly, as shown in Eq. 1, we present the LASSO in the safe-screen style (Xiang et al., 2016), which allows a subset of features receiving zero weights to be identified before minimizing the loss function, as shown in Theorem 1.

$$
\begin{aligned}
&\min_w L(w, T_1, T_2, ..., T_{N_T}) = \min_w L_s(w, T_1, T_2, ..., T_{N_T}) + \sum_{i=1}^{N_T}\sum_{j=1}^{l_{T_i}} |w_j'^{T_i}|, \\
&\text{s.t.} \quad L_s(w, T_1, T_2, ..., T_{N_T}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}(y - \sum_{i=1}^{N_T}\sum_{j=1}^{l_{T_i}} w_j'^{T_i} x_j^{T_i})^2 \\
&w_j'^{T_i} = \begin{cases} w_j^{T_i} & j \in w_\downarrow^{T_i}, \\ 0 & \text{otherwise}, \end{cases} = \begin{cases} w_j^{T_i} & j \in w_\downarrow'^{T_i} \cup w_\downarrow''^{T_i}, \\ 0 & \text{otherwise}, \end{cases} \\
&w_\downarrow'^{T_i} = \{i | \mathbb{E}x_j^{T_i} y + c_F\sqrt{\mathbb{E}x_j^{T_i}} > \lambda, j \in [l_{T_i}]\}, \\
&w_\downarrow''^{T_i} = \{j | pa(n_{T_i,j'}^l) = pa(n_{T_i,j}^l), j' \in w_\downarrow'^{T_i}, j \in [l_{T_i}]\},
\end{aligned}
\tag{1}
$$

where w is the set of all trainable weights in L, and λ is a positive real number measuring the degree of regularization, c_F is a positive real number, and $n_{T_i,j}^l$ denotes the j^{th} leaf node of T_i and $pa(\cdot)$ the parent node. To avoid $w = 0$, we have $\lambda \in [0, \lambda_{max}]$, where $\lambda_{max} = max_{i,j}(|\mathbb{E}x_j^{T_i} y|)$.

The safe-screen LASSO shown in Eq. 1 can be regarded as a metric to evaluate the quality of interactions transformed by the set of DTs. By solving $\arg\min_{T_1,...,T_{N_T}} min_w L(w, T_1, T_2, ..., T_{N_T})$, the optimal set of DTs (or interaction transformers) can be identified. In Eq. 1, the index set $w_\downarrow^{T_i}$ is calculated, and all weights with indices that do not belong to $w_\downarrow^{T_i}$ are set to zeros in L prior to the minimization of L.

Assumption 1. $0 < \frac{\lambda}{\lambda_{max}} \leq (1 - \frac{c_F}{\sqrt{\mathbb{E}_{(x,y)\sim\mathcal{D}} y^2}}) < 1$.

Theorem 1. *Under assumption 1, let $\{w^{*,T_1}, w^{*,T_2}, ..., w^{*,T_{N_T}}\}$ be the solution of the corresponding LASSO problem of Eq. 1, if $j \notin w_\downarrow^{T_i}$, $w_j^{*,T_i} = 0$.*

Remark 1. *For each interaction feature $x_j^{T_i}$, if $y \in \{0, 1\}$, the left-hand side of the screen criterion in $w_\downarrow'^{T_i}$ is a convex combination between $x_j^{T_i}$'s true positive rate (TPR) and frequency. If $x_j^{T_i}$ has low TPR and frequency, its weight is more likely to equal zero in Eq. 1.*

Remark 2. *Let T_i' be the DT after pruning all T_i's leaf nodes whose indices are not included in $w_\downarrow^{T_i}$, and removing parent nodes of these leaf nodes. By including $w_\downarrow''^{T_i}$ into $w_\downarrow^{T_i}$, $\forall i \in [1, 2, ..., N_T]$, $x^{T_i'}$ will always be a one-hot encoding vector, which can simplify the following problem formulation.*

Even though the safe-screen LASSO reduce the simplify the optimization problem by discarding a set of zero-weighted features before minimizing the loss function, the optimization problem in Eq. 1 does not have a closed-form solution, which requires a computationally expensive numerical method to minimize the loss function. To further simplify the calculation, after setting weights whose indices are not included in $w_\downarrow^{T_i}$ as zero, we ignore the $l1$ penalty, and relax the the minimization of L into minimizing L_s's upper bound function $\overline{L_s}$, as shown in Eq. 2.

$$
\min_w \overline{L_s}(w, T_1, T_2, ..., T_{N_T}) = \min_w \frac{1}{N_T}\sum_{i=1}^{N_T} \overline{L_s}^{T_i}(w^{T_i}) = \min_w \frac{1}{N_T}\sum_{i=1}^{N_T} \mathbb{E}_{(x,y)\sim\mathcal{D}}(y - \sum_{j=1}^{l_{T_i}} w_j'^{T_i} x_j^{T_i})^2 \tag{2}
$$

It is possible to easily minimize $\overline{L_s}^{T_i}, \forall i \in \{1, 2, ..., N_T\}$ independently in Eq. 2, as shown in Theorem 2.

Theorem 2. *Let* $w^{*,T_i} = \arg\min_{w^{T_i}} \overline{L_s}^{T_i}(w^{T_i})$ *and* $L^*_{T_i} = min_{w^{T_i}} \overline{L_s}^{T_i}(w^{T_i})$. *With the event* $\mathcal{E}_{x,T_i,j} = \{x$ *is assigned to* T_i*'s* j^{th} *leaf node*$\}$, *we have*

$$L^*_{T_i} = \sum_{j \in w^{T_i}_{\downarrow}} P(\mathcal{E}_{x,T_i,j}) \mathbb{E}\big[(y - \mathbb{E}\big[y|\mathcal{E}_{x,T_i,j}\big])^2 | \mathcal{E}_{x,T_i,j}\big] + \sum_{j \in [l_{T_i}] \setminus w^{T_i}_{\downarrow}} P(\mathcal{E}_{x,T_i,j}) \mathbb{E}\big[y^2 | \mathcal{E}_{x,T_i,j}\big]. \tag{3}$$

While the minimum loss of DT can be calculated with Eq. 3, it would be infeasible to choose the best one from a large number of candidates. Accordingly, a rule is proposed to reduce the number of candidate DTs.

Theorem 3. *Let* $\mathcal{T}_i$ *be the candidate DT set of the* i^{th} *DT* T_i. *With* $\mathcal{T}'_i = \{T | T \in \mathcal{T}_i \ \ \& \ \ \{1, 2, ..., l_{T_i}\} = w^{T_i}_{\downarrow}\}$, *we build* $\mathcal{T}^-_i = \{T | T \in \mathcal{T}'_i \ \ \& \ \ \mathcal{T}'_i \setminus \{T\}$ *doesn't contain* T*'s sub-tree*$\}$. $\forall T \in \mathcal{T}_i$, $\exists T' \in \mathcal{T}^-_i$ *with* $L^*_{T'} \leq L^*_T$.

Remark 3. *The proof of Theorem 3 can be made by considering two factors of Eq. 3: 1) with* $T \in \mathcal{T}$, *when we prune* T*'s nodes with indices that do not belong to* $w^T_{\downarrow}$ *and remove their parent nodes, the resultant* T' *has a lower loss; 2) If* T *can be expanded to* T'' *by splitting one of* T*'s leaf node with a specific feature in* x, *and all resultant new leaf nodes belong to* $w^{T''}_{\downarrow}$, *we have* $L^*_{T''} \leq L^*_T$.

Remark 4. *For* $\forall T \in \mathcal{T}^-_i$ *and* $\forall (x, y) \in \mathcal{D}$, *based on the analysis in Remark 2,* x *can be transferred into a one-hot encoding feature by* T.

With $\mathcal{T}^-_i$ defined in Theorem 3, the optimal DT can be identified from $\mathcal{T}^-_i$ instead of $\mathcal{T}_i$, which leaves the second term of Eq. 3 at zero. Therefore, for $\forall i \in \{1, 2, ..., N_T\}$, the optimal DT can be obtained by solving Eq. 4.

$$T^*_i = \arg\min_{T \in \mathcal{T}^-_i} \sum_{j=1}^{l_{T_i}} P(\mathcal{E}_{x,T_i,j}) \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[(y - \mathbb{E}\big[y|\mathcal{E}_{x,T_i,j}\big])^2 | \mathcal{E}_{x,T_i,j}\big]. \tag{4}$$

For the sake of simplicity, the subscripts of T^*_i, T_i and $\mathcal{T}_i$ are omitted.

2.2 Identifying the optimal DT with the divide-and-conquer approach

To efficiently obtain the optimal DT in Eq. 4, we introduce a divide-and-conquer approach to identify T^*, which is more time- and space- efficient compared with separately computing loss for each of DTs in $\mathcal{T}^-$. Before presenting the approach, we define a data structure FDT (Feature Involved DT) in Definition 2 for storing DTs in $\mathcal{T}$.

Definition 2 (FDT). *FDT is a DT (Definition 1) except that each of its data nodes (*n^d*) contains all candidate feature nodes, whose indices differ from those of all feature nodes within the path from FDT's root node to* n^d.

FDT is a better space-saving alternative than storing DTs separately since only one path is saved in the FDT if multiple DTs share the same path from the root node to any other node. From FDT, one can derive DT by selecting at most one feature node for each of FDT's data nodes, as shown in Fig. 1b. Although the fully expanded FDT is a huge tree, most of its nodes are not associated with DTs in $\mathcal{T}^-$, and thus can be ignored in searching the optimal DT shown in Eq. 4. Theorem 4 proposes a pruning strategy to reduce the size of the FDT, and we demonstrate that each of the DTs in $\mathcal{T}^-$ can be obtained from the pruned FDT.

Theorem 4. *Let* T^P *be the FDT after pruning its feature nodes whose child data nodes all satisfy the criterion set forth in Eq. 5 , and* $\mathcal{T}^P$ *be the set of all candidate DTs obtained from* T^P *by selecting* ***one and only one*** *feature node for each of* T^P*'s internal data nodes. We have* $\mathcal{T}^P = \mathcal{T}^-$.

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y 1_{(x,y)\in S_{n^d}}\big] + c_F \sqrt{P((x,y) \in S_{n^d})} < \lambda, \tag{5}$$

where $P((x,y) \in S_{n^d})$ *and* $\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y 1_{(x,y)\in S_{n^d}}\big]$ *can be further decomposed into Eq. 6 according to Bayes' theorem.*

$$\begin{aligned} P((x,y) \in S_{n^d}) &= \prod_{n^{q'}_{i,j} \in e_{n^d} \setminus n_r} P(x_i = j | (x,y) \in S_{pa(pa(n^{q'}_{i,j}))}), \\ \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y 1_{(x,y)\in S_{n^d}}\big] &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y | (x,y) \in S_{n^d}\big] P((x,y) \in S_{n^d}), \end{aligned} \tag{6}$$

where e_{n^d} is the node set containing all data nodes along the path from the FDT's root node to n^d, and $pa(pa(n^{d'}_{i,j}))$ indicates $n^{d'}_{i,j}$'s grandparent node.

The optimal DT can be identified in T^P via the divide-and-conquer approach. First, we denote T^P's sub-tree rooted in n^d as $T^P_{n^d}$. DTs can be obtained from $T^P_{n^d}$ by selecting one and only one child feature nodes for each of $T^P_{n^d}$'s data node. Let $\mathcal{T}^P_{n^d}$ be the set of all candidate DTs obtained from $T^P_{n^d}$. By replacing $\mathcal{D}$ and $\mathcal{T}^-$ in Eq. 4 with S_{n^d} and $\mathcal{T}^P_{n^d}$, we create a sub-problem $\mathcal{P}_{n^d}$ of Eq. 4, identifying the DT with the minimum loss from $\mathcal{T}^P_{n^d}$. It's easy to prove that the optimal DT identified in $\mathcal{P}_{n^{d'}}$ is a sub-tree of that identified in $\mathcal{P}_{n^d}$ if $T^P_{n^{d'}}$ is $T^P_{n^d}$'s sub-tree. Therefore, if n^d is T^P's internal data node, $L^*_{n^d}$ can be calculated based on the solutions of other sub-problems, as shown in Eq. 7.

$$L^*_{n^d} = \min_{n^f \in c(n^d)} L^*_{n^f} = \min_{n^f \in c(n^d)} \sum_{n^d_{i,j} \in c(n^f)} P(x_i = j | (x, y) \in S_{n^d}) L^*_{n^d_{i,j}}. \tag{7}$$

If n^d is T^P's leaf node, $T^P_{n^d}$ has only one node (i.e. n^d). Therefore, we simply have $L^*_{n^d} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y|(x, y) \in S_{n^d}\big])^2 | (x, y) \in S_{n^d}]$.

By recursively calculating $L^*_{n^d}$ for each of the data nodes in T^P, we can finally get the minimum loss in Eq. 4 by calculating $L^*_{n_r}$, where n_r is the root node of T^P. For each of the internal data nodes in T^P, only one child feature node is selected, which enables us to determine the corresponding optimal DT with $L^*_{n_r}$.

3 The Proposed MCTs-IS Method

Before pruning T^P and obtaining the optimal DT via Eqs 5 and 7, parameters in Eqs. 6 and 7 should be obtained first. Therefore, estimators are built to estimate these parameters on the basis of $(x, y) \in \mathcal{D}$. Since estimating all unknown parameters with a big $\mathcal{D}$ is computationally intractable because of the complexity of FDT, we propose MCTs-IS to calculate these estimators.

MCTs-IS estimates FDT by constructing FDT iteratively from scratch. Let the constructed FDT in the t^{th} iteration be $\hat{T}_t$. Estimators $\hat{p}_{n^d_{i,j},t}$, $\hat{\mu}_{n^d,t}$ and $\hat{\sigma}_{n^f,t}$ are saved in data and feature nodes of $\hat{T}_t$, and used to estimate $P(x_i = j|(x, y) \in S_{pa(pa(n^d_{i,j}))})$, $\mathbb{E}\big[y|(x, y) \in S_{n^d}\big]$ and $L^*_{n^f}$ in Eqs 6 and 7.

At the beginning of the MCTs-IS, $\hat{T}_t$ only has a root node n_r. In each iteration, three steps, *selection*, *backpropagation* and *expansion*, are conducted in MCTs-IS to select a set of data and feature nodes from $\hat{T}_t$, update estimators of parameters in selected data nodes with the sampled data (x_t, y_t), and adding new feature and data nodes to $\hat{T}_t$. By repeatedly adding new nodes to $\hat{T}_t$, each of T^P's node can finally be found in $\hat{T}_t$. The pipeline of MCTs-IS is shown in Algorihtms 1.

1) Selection. In this step, a set of $\hat{T}_t$'s data nodes will be selected, which starts from its root node. Each selected data node n^d will group its child feature nodes that do not meet the pruning guidelines presented in Theorem 4 into a selectable feature set $l(n^d)$, and then the feature node with the (potentially) lowest estimated $L^*_{n^f}$ will be selected from $l(n^d)$ according to the node selection policy $\pi_t(\cdot)$ as described later. To estimate whether a $\hat{T}_t$'s feature node n^f meets the pruning criterion in Theorem 4, we replace unknown parameters of the left-hand side of Eq.5 with estimators to build $s_{n^d} = \prod_{n^{d'}_{i,j} \in e_{n^d} \setminus n_r} \hat{p}_{n^{d'}_{i,j},t} + c_F \sqrt{\hat{\mu}_{n^d,t} \prod_{n^{d'}_{i,j} \in e_{n^d} \setminus n_r} \hat{p}_{n^{d'}_{i,j},t}}$. Following Theorem 4, if for $\forall n^d \in c(n^f)$, $s_{n^d} \leq \lambda$, n^f meets the estimated pruning criterion, and thus belongs to $l(pa(n^f))$. For each selected feature node n^f, n^f's child data node with $(x_t, y_t) \in S_{n^d}$ is selected. As shown in line 18 of the Algorithm 1, the selection procedure will be conducted repeatedly until the agent selects a $\hat{T}_t$'s leaf node, or a $\hat{T}_t$'s data node n^d with $l(n^d) = \emptyset$. The main challenge in selecting feature node is to maintain the balance between exploiting fully estimated feature node and exploring those with limited estimation, i.e. *exploitation vs. exploration*. Accordingly, a feature node selection policy $\pi_t(\cdot) : n^d \to n^f$ is proposed in Eq. 8.

$$\pi_t(n^d) = \arg \min_{n^f \in l(n^d)} \hat{\sigma}_{n^f,t} - C_p \sqrt{\frac{\log \sum_{n^{f'} \in l(n^d)} V_{n^{f'},t}}{V_{n^f,t}}}, \tag{8}$$

Algorithm 1: MCT-IS

1: **Procedure** MCT-IS
2: **Input:** $\hat{T}_t$'s root n_r, the total number of rounds t_{max}.
3: $t = 1$
4: **repeat**
5: Sampling a data (x_t, y_t) from the dataset
6: Search(n_r,$\{n_r\}$,1,1,t,(x_t, y_t))
7: $t = t + 1$
8: **until** $t > t_{max}$
9:
10: **Function** Search
11: **Input:** $\hat{T}_t$'s data node n^d, the set SN_t, pruning strategy estimator p_{yb} and p_b, number of iteration t and sampled data (x_t, y_t).
12: **//Expansion**
13: If n^d has no child node, add feature nodes to n^d, and add data nodes to all added feature nodes
14: **if** $c(n^d) = \emptyset$ or $l(n^d) = \emptyset$ **then**
15: **Return** SN_t
16: **else**
17: **//Selection**
18: $n_f = \pi_t(n^d)$, $n^{d'} \leftarrow n^f$'s child data node $n^{d'}$ with $(x_t, y_t) \in S_{n^{d'}}$
19: $SN_t \leftarrow SN_t \cup \{n^{d'}\}$
20: $SN_t = Search(n^{d'}, d + 1, p_{yb}, p_b, t, (x_t, y_t))$
21: **//Back-propogation**
22: **for** $n^{f'} \in c(n^{d'}) \backslash l(n^{d'}) \cup \{n^f\}$ **do**
23: $n^{d''} \leftarrow n^{f'}$'s child data node $n^{d''}$ with $(x_t, y_t) \in S_{n^{d''}}$
24: Update $V_{n^{d''},t+1}$,$B_{n^{d''},t+1}$, $\hat{p}_{n^{d''},t+1}$ and $\hat{\mu}_{n^{d''},t+1}$ following section 3.2
25: $s_{n^{d''}} \leftarrow p_{yb}\hat{\mu}_{n^{d''},t+1} + c_F\sqrt{p_b\hat{p}_{n^{d''},t+1}}$
26: **end for**
27: Update $\hat{\sigma}_{n^f,t+1}$ following Eq. 9, $V_{n^f,t+1} \leftarrow V_{n^f,t} + 1$
28: **end if**
29: **Return** SN_t

where $V_{n^f,t}$ measures how many times n^f has been selected so far, $\hat{\sigma}_{n^f,t}$ is the estimator of $L^*_{n^f}$, whose definition will be given in the *backpropagation* subsection.

Remark 5. *The second term of Eq. 8 considers the uncertainty of estimation as opposed to greedily selecting the node with the lowest $\hat{\sigma}_{n^f,t}$. Specifically, the more data are used to estimate the parameter, the less uncertainty in estimation we may encounter. In this case, a feature node estimated by few data can be selected even though it has a higher $\hat{\sigma}_{n^f,t}$.*

2) Backpropagation. Let SN_t be the set of selected data nodes in the *selection* step, and we further denote $SN_{t,i}$ as the i^{th} selected data node. In the step of *back-propagation*, the estimators within SN_t will be updated by the sampled (x_t, y_t). Note that the memory cost in this step is small since all estimators can be updated incrementally. For each $n^d_{i,j} \in SN_t$, we have ${}_{n^d_{i,j},t+1} = B_{n^d_{i,j},t} + y_t$, $V_{n^d_{i,j},t+1} = V_{n^d_{i,j},t} + 1$, $\hat{p}_{n^d_{i,j},t+1} = \frac{V_{n^d_{i,j},t+1}}{\sum_{j'=1}^{K_i} V_{n^d_{i,j'},t+1}}$ and $\hat{\mu}_{n^d,t+1} = \frac{B_{n^d,t+1}}{V_{n^d,t+1}}$. Given a $\hat{T}$'s feature node n^f, if $n^f \notin l(pa(n^f))$, and estimators calculating $l(pa(n^f))$ are not updated in the following iterations of MCTs-IS, n^f will no longer be selected even though n^f is mistakenly excluded from $l(pa(n^f))$. Therefore, if $n^f \notin l(pa(n^f))$, we additionally update $V_{n^d,t+1}$, $B_{n^d,t+1}$, $\hat{p}_{n^d_{i,j},t}$, $\hat{\mu}_{n^d_{i,j},t}$ and s_{n^d} in each of n^f's child node n^d, as shown in line 22-26 of the Algorithm 1. By updating these nodes, n^f might be included in $l(pa(n^f))$ again, and then selected by MCTs-IS.

After updating $\hat{p}_{n^d_{i,j},t}$ and $\hat{\mu}_{n^d_{i,j},t}$, for $\forall SN_{t,i} \in SN_t \backslash n_r$, $\hat{\sigma}_{pa(SN_{t,i}),t}$ will be updated with Eq. 9.

$$\begin{aligned} \hat{\sigma}_{pa(SN_{t,i}),t+1} &= \frac{V_{pa(SN_{t,i}),t}\hat{\sigma}_{pa(SN_{t,i}),t} + L_{pa(SN_{t,i}),t}}{V_{pa(SN_{t,i}),t}+1}, \\ s.t.\ L_{pa(SN_{t,i}),t} &= \sum_{j=i}^{|SN_t|} w_{SN_t,i,j}(y_t - \hat{\mu}_{SN_{t,i},t+1})^2 \\ w_{SN_t,i,j} &= \frac{(\gamma_{SN_t})^j}{\sum_{j'=i}^{|SN_t|}(\gamma_{SN_t})^{j'}}, \quad \gamma_{SN_t} = \frac{\upsilon + V_{SN_{t,|SN_t|},t+1}}{\kappa}, \end{aligned} \tag{9}$$

where $\sum_{j=i}^{|SN_t|} w_{SN_t,i,j} = 1$, $\upsilon, \kappa \in \mathbb{R}^+$, and $\frac{\upsilon}{\kappa} \in [0,1]$.

Remark 6. *$L_{pa(SN_{t,i}),t}$ in Eq. 9 is a weighted (measured by $w_{SN_t,i,j}$) mean square errors of data nodes in $\{SN_{t,j} | j \in [i, |SN_t|]\}$. When the visit time of the deepest node in SN_t (i.e. $SN_{t,|SN_t|}$) is small ($\gamma_{SN_t} < 1$), the square errors of shallow nodes carry a larger weight than deep nodes and it produces a stable heuristic information which guides the following iterations, since shallow nodes are estimated by more samples than deep ones. As the visit time of the deepest node in SN_t is large enough ($\gamma_{SN_t} > 1$), the deepest node in SN_t has the largest weight. In Appendix B, for each of T^P's leaf nodes' parent feature nodes, we show that $\lim_{t\to\infty} \mathbb{E}[\hat{\sigma}_{n^f,t}] = L^*_{n^f}$, in Lemma 3.*

After updating $\hat{\sigma}_{pa(SN_{t,i}),t}$, for each parent node of data node $SN_{t,i}$ in SN_t, we have $V_{pa(SN_{t,i}),t} = V_{pa(SN_{t,i}),t-1} + 1$.

3) Expansion. At the step of *expansion*, if the deepest node in SN_t has no child nodes, all of its candidate child feature nodes and the resulted data nodes will be appended to it, as shown in line 13 of Algorithm 1. When a new feature node n^f is added to $\hat{T}_t$, $\hat{\sigma}_{n^f,t}$ is set to 0; when a new data node n^d is added, the initial values of $V_{n^d,t}$, $B_{n^d,t}$ and s_{n^d} are set to 1, 0 and 1, respectively.

4 Theoretical Analysis

To the best of our knowledge, we are the first work to formulate the relationship between the integration selection and the MCTs, and in Theorem 5, we show that the probability obtaining the optimal DT in MCTs-IS converges to 1. The proof of Theorem 5 can be found in Appendix A.

Theorem 5. *In the t^{th} iteration of MCTs-IS, let $\hat{T}^P_t$ be the FDT after pruning all $\hat{T}_t$'s feature nodes n^f with $n^f \notin l(pa(n^f))$. For each of $\hat{T}^P_t$'s leaf nodes' parent feature node $n^{f'}$, if $L_{n^{f'}}$ is estimated by $\hat{\sigma}_{n^{f'},t}$, and for each $\hat{T}^P_t$'s internal data node n^d, $L^*_{n^d}$ is calculated on the basis of the estimated $L_{n^{f'}}$, a DT $\hat{T}^*_t$ can be obtained from $\hat{T}^P_t$, and we have $\lim_{t\to\infty} P(\hat{T}^*_t = T^*) = 1$.*

5 Experiments

In this section, extensive experiments were performed to confirm the performance of interactions identified by MCTs-IS in boosting the predictive results of machine learning models.

Datasets. We evaluated MCTs-IS and baseline algorithms on Avazu [1] and TYGEM [2]. Avazu is a Click-through rate (CTR) dataset widely used for bench-marking interaction identification algorithms. We processed Avazu dataset following Sun et al. (2021)'s pipeline. After data processing, each of ∼40M samples in Avazu dataset has 23 categorical fields, which can be further transferred into ∼1.54M binary features. Although interactions exists in the dataset of Avazu, most of its features are anonymized, which makes it hard to interpret the identified interactions. Therefore, we further evaluated algorithms with records of matches played on the online Go game platform called TYGEM. High order interactions are important in playing the game of Go, since moves of both real and AI Go players are heavily relied on interactions among previous moves (also called Shapes and Connections). Following the data preprocessing pipeline in Appendex B, we extract a binary classification dataset from TYGEM, which has $\sim 1.9M$ records, and each record has 48 ternary features.

[1] https://www.kaggle.com/c/avazu-ctr-prediction/data

[2] https://github.com/yenw/computer-go-dataset

Dataset	Model	Interaction selection method					
		N/A		+GBDT		+MCTs-IS(ours)	
		Test AUC	Test Loss	Test AUC	Test Loss	Test AUC	Test Loss
Avazu	LR	0.750	0.396	0.756	0.394	**0.765**	**0.388**
	FM	0.770	0.386	0.768	0.387	**0.771**	**0.385**
	FmFM	0.778	**0.382**	0.773	0.392	**0.779**	0.383
TYGEM	LR	0.658	0.556	0.786	0.455	**0.794**	**0.449**
	FM	0.795	0.468	0.788	0.452	**0.797**	**0.446**
	FmFM	**0.8088**	0.457	0.786	0.455	0.8087	**0.439**

Table 1: Performance comparison

Both Avazu and TYGEM datasets were randomly split into 8:1:1 as the training set, validation set, and test set.

Baseline Algorithms. After transferring features in Avazu and TYGEM into high-order interactions with MCTs-IS and the Gradient Boosting Decision Tree (GBDT) algorithm, a widely used DT-based interaction identification method, we evaluated the performance of these high-order interactions in boosting the AUC and log loss of shallow models, i.e. Logistic regression (LR), FM and Field-matrixed Factorization Machines (FmFM), the state-of-the-art shallow method in predicting the Avazu dataset. Among these baseline algorithms, GBDT is implemented with LightGBM (Ke et al., 2017), a gradient boosting framework that uses tree based learning algorithms, while LR, FM and FmFM were trained and tested with the open-source training code of FmFM (Sun et al., 2021) implemented on the basis of the TensorFlow(Abadi et al., 2016).

Experimental Methods. We randomly sampled 1M records from the training set to train both the MCTs-IS and the GBDT. With the fitted DT-based methods, all features in the train, valid and test sets were transferred into interactions. After that, LR, FM and FmFM were used to predict labels with these interactions under different hyper-parameter setting (more details can be found in Appendix B) and optimizers (SGD with momentum and Adam). After selecting the hyper-parameter setting with the highest AUC (Area Under the ROC Curve) on the validation set, the AUC and the log loss on the test set were used to evaluate performances of baseline algorithms. We implement all the algorithms on a server equipped with Intel Xeon Gold 6150 2.7GHz CPU, 192GB RAM, and an NVIDIA Tesla V100 GPU. The result is shown in Table 5.

Result and Discussion. In Table 5, we have four observations. 1) Compared with reported results in the paper proposing FmFM, we got better results in the Avazu dataset because instead of Adam, we use SGD with momentum to solve FM and FmFM, which usually have a better performance than Adam in non-convex optimization. 2) LR+MCTs-IS, FM+MCTs-IS and FmFM+MCTs-IS had better predictive results than those with GBDT. 3) Although high-order interactions can be identified by MCTs-IS and GBDT, they failed to boost the performance of FM and FmFM, which considered all pairwise interactions. It indicates that the second-order feature interactions plays more important roles in Avazu dataset than high-order interactions since only a subset of two-way interactions can be identified by DTs. 4) In the Go game dataset TYGEM, high order interactions are critical for both real and AI players to learn how to play the Go game. Therefore, FmFM+MCTs-IS achieved much lower loss compared with FmFM, which means that by learning knowledge patterns in Go games visualized in Appendix B, FmFM got more confidence in the label prediction.

CONCLUSIONS AND FUTURE WORK

In this work, we propose MCTs-IS to identify high-way interactions in an online learning manner based on the framework of MCTs. A new pruning strategy is derived based on the safe-screen LASSO modeling the interaction selection problem to reduce the search space. We also provide the theoretical analysis of the convergence on identifying interactions with the MCTs framework. In the future, as an extensible framework, the MCTs in the MCTs-IS may be enhanced by various technologies, including parallel computing, expert knowledge, and DNNs. Additionally, we are interested in finding a low dimensional representation of the identified sparse high-way interactions, so that they can be applied to deep learning models without over-fitting.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016.

Junmei Bao, Yangguang Ji, Yonghui Yang, Le Wu, and Ruiji Fu. Resfusion: A residual learning based fusion framework for ctr prediction. In China Conference on Information Retrieval, pp. 29–41. Springer, 2020.

Jacob Bien, Jonathan Taylor, and Robert Tibshirani. A lasso for hierarchical interactions. Annals of statistics, 41(3):1111, 2013.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems, pp. 7–10, 2016.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247, 2017.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, pp. 1–9, 2014.

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.

Levente Kocsis, Csaba Szepesvári, and Jan Willemson. Improved monte-carlo search. Univ. Tartu, Estonia, Tech. Rep, 1, 2006.

Marek Kretowski and Marek Grzes. Global learning of decision trees by an evolutionary algorithm. In Information Processing and Security Systems, pp. 401–410. Springer, 2005.

Hyafil Laurent and Ronald L Rivest. Constructing optimal binary decision trees is np-complete. Information processing letters, 5(1):15–17, 1976.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1754–1763, 2018.

Michael Lim and Trevor Hastie. Learning interactions via hierarchical group-lasso regularization. Journal of Computational and Graphical Statistics, 24(3):627–654, 2015.

Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2636–2645, 2020.

Yuanfei Luo, Mengshuo Wang, Hao Zhou, Quanming Yao, Wei-Wei Tu, Yuqiang Chen, Wenyuan Dai, and Qiang Yang. Autocross: Automatic feature crossing for tabular data in real-world applications. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1936–1945, 2019.

Harshit Pande. Field-embedded factorization machines for click-through rate prediction. arXiv preprint arXiv:2009.09931, 2020.

Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1149–1154. IEEE, 2016.

Steffen Rendle. Factorization machines. In 2010 IEEE International conference on data mining, pp. 995–1000. IEEE, 2010.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484–489, 2016.

Daria Sorokina, Rich Caruana, Mirek Riedewald, and Daniel Fink. Detecting statistical interactions with additive groves of trees. In Proceedings of the 25th international conference on Machine learning, pp. 1000–1007, 2008.

Yang Sun, Junwei Pan, Alex Zhang, and Aaron Flores. Fm2: Field-matrixed factorization machines for recommender systems. In Proceedings of the Web Conference 2021, pp. 2828–2837, 2021.

Zhulin Tao, Xiang Wang, Xiangnan He, Xianglin Huang, and Tat-Seng Chua. Hoafm: a high-order attentive factorization machine for ctr prediction. Information Processing & Management, 57(6): 102076, 2020.

Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. arXiv preprint arXiv:1705.04977, 2017.

Michael Tsang, Dehua Cheng, Hanpeng Liu, Xue Feng, Eric Zhou, and Yan Liu. Feature interaction interpretability: A case for explaining ad-recommendation systems via neural interaction detection. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=BkgnhTEtDS.

Gregory Vaughan, Robert Aseltine, Kun Chen, and Jun Yan. Efficient interaction selection for clustered data via stagewise generalized estimating equations. Statistics in Medicine, 39(22): 2855–2868, 2020.

Zhen James Xiang, Yun Wang, and Peter J Ramadge. Screening tests for lasso problems. IEEE transactions on pattern analysis and machine intelligence, 39(5):1008–1027, 2016.

A APPENDIX A

Theorem 1. *Under assumption 1, let* $\{w^{*,T_1}, w^{*,T_2}, ..., w^{*,T_{N_T}}\}$ *be the solution of the corresponding LASSO problem of Eq. 1, if* $j \notin w_{\downarrow}^{T_i}$, $w_j^{*,T_i} = 0$.

Proof. Suppose that $\mathcal{D}$ has N samples, and let (x_t, y_t) be the t^{th} sample of $\mathcal{D}$. We define the loss function of the LASSO as

$$\min_w \sum_{t=1}^{N}(y_t - \sum_{i=1}^{N_T}\sum_{j=1}^{l_{T_i}} w_j^{T_i} x_{t,j}^{T_i})^2 + N\lambda\|w\|_1, \tag{10}$$

where $\lambda \leq \lambda_{max} = max_{i,j}(\frac{\sum_{t=1}^{N} x_{t,j}^{T_i} y_t}{N})$. Following the proof of Theorem 2 in section 4.2 of the safe-screen LASSO (Xiang et al., 2016), given $i \in [N_T]$ and $j \in [l_{T_i}]$, if

$$\frac{1}{N}\sum_{t=1}^{N} x_{t,j}^{T_i} y_t + (1 - \frac{\lambda}{\lambda_{max}})\sqrt{\frac{\sum_{t=1}^{N} y_t^2}{N}\frac{\sum_{t=1}^{N}(x_{t,j}^{T_i})^2}{N}} \leq \lambda, \tag{11}$$

we have $w_j^{T_i} = 0$. Then let $N \to \infty$, and based on Assumption 1, Eq. 11 can be rewritten as

$$\mathbb{E}_{(x,y)\sim\mathcal{D}} x_j^{T_i} y + c_F\sqrt{\mathbb{E}_{(x,y)\sim\mathcal{D}} x_j^{T_i}} \leq \lambda,$$

which is the screen criterion in the set of $w_{\downarrow}^{\prime T_i}$. Since weights of features whose indices are not included in $w_{\downarrow}^{\prime T_i}$ equal zero, and $w_{\downarrow}^{\prime T_i}$ is a subset of $w_{\downarrow}^{T_i}$, if $j \notin w_{\downarrow}^{T_i}$, $w_j^{*,T_i} = 0$, which finish the proof. □

Theorem 2. *Let* $w^{*,T_i} = \arg\min_{w^{T_i}} \overline{L_s}^{T_i}(w^{T_i})$ *and* $L_{T_i}^* = min_{w^{T_i}} \overline{L_s}^{T_i}(w^{T_i})$. *With the event* $\mathcal{E}_{x,T_i,j} = \{x$ *is assigned to* T_i*'s* j^{th} *leaf node*$\}$, *we have*

$$L_{T_i}^* = \sum_{j\in w_{\downarrow}^{T_i}} P(\mathcal{E}_{x,T_i,j})\mathbb{E}\big[(y - \mathbb{E}\big[y|\mathcal{E}_{x,T_i,j}\big])^2|\mathcal{E}_{x,T_i,j}\big] + \sum_{j\in[l_{T_i}]\setminus w_{\downarrow}^{T_i}} P(\mathcal{E}_{x,T_i,j})\mathbb{E}\big[y^2|\mathcal{E}_{x,T_i,j}\big]. \tag{3}$$

Proof. Since for any $j \notin w_{\downarrow}^{T_i}$, $w_j^{\prime T_i} = 0$. Therefore, $min_{w^{T_i}} \overline{L_s}^{T_i}(w^{T_i})$ can be written as

$$\begin{aligned} &min_{w^{T_i}} \overline{L_s}^{T_i}(w^{T_i}) \\ =&min_{w^{T_i}} \sum_{j\in w_{\downarrow}^{T_i}} P(\mathcal{E}_{x,T_i,j})\mathbb{E}\big[(y - w_j^{T_i})^2|\mathcal{E}_{x,T_i,j}\big] + \sum_{j\in[l_{T_i}]\setminus w_{\downarrow}^{T_i}} P(\mathcal{E}_{x,T_i,j})\mathbb{E}\big[y^2|\mathcal{E}_{x,T_i,j}\big]. \end{aligned} \tag{12}$$

The first term of Eq. 12 has a closed-form solution

$$w_j^{T_i} = \mathbb{E}\big[y|\mathcal{E}_{x,T_i,j}\big]. \tag{13}$$

Substitute Eq. 13 into Eq 12, we finally get

$$\begin{aligned} L_{T_i}^* =&min_{w^{T_i}} \overline{L_s}^{T_i}(w^{T_i}) \\ =& \sum_{j\in w_{\downarrow}^{T_i}} P(\mathcal{E}_{x,T_i,j})\mathbb{E}\big[(y - \mathbb{E}\big[y|\mathcal{E}_{x,T_i,j}\big])^2|\mathcal{E}_{x,T_i,j}\big] + \sum_{j\in[l_{T_i}]\setminus w_{\downarrow}^{T_i}} P(\mathcal{E}_{x,T_i,j})\mathbb{E}\big[y^2|\mathcal{E}_{x,T_i,j}\big]. \end{aligned}$$

□

Theorem 3. *Let* $\mathcal{T}_i$ *be the candidate DT set of the* i^{th} *DT* T_i. *With* $\mathcal{T}_i' = \{T|T \in \mathcal{T}_i \ \ \& \ \ \{1, 2, ..., l_{T_i}\} = w_{\downarrow}^{T_i}\}$, *we build* $\mathcal{T}_i^- = \{T|T \in \mathcal{T}_i' \ \ \& \ \ \mathcal{T}_i'\setminus\{T\}$ *doesn't contain* T*'s sub-tree*$\}$. $\forall T \in \mathcal{T}_i$, $\exists T' \in \mathcal{T}_i^-$ *with* $L_{T'}^* \leq L_T^*$.

Proof. For any $T \in \mathcal{T}_i$ with $w_{\downarrow}^T \subset \{1, 2, ..., l_T\}$, we can repeatedly pruning T's feature nodes all of whose child nodes' indices are not included in $w_{\downarrow}^T$ until we get T' with $w_{\downarrow}^{T'} = \{1, 2, ..., l_{T'}\}$. According to factor 1 in Remark 3, by pruning T's leaf nodes whose indices are not included in $w_{\downarrow}^T$, the resultant DT has a lower loss. Therefore, we can prove that for any $T \in \mathcal{T}_i$, there exists a DT in $\mathcal{T}_i'$, whose loss is smaller than that of T. Then according to factor 2 in Remark 3, for any $T \in \mathcal{T}_i'$, if T is the sub-tree of any DT in $\mathcal{T}_i' \backslash \{T\}$, it has larger loss, which concludes the proof. □

Theorem 4. *Let T^P be the FDT after pruning its feature nodes whose child data nodes all satisfy the criterion set forth in Eq. 5 , and $\mathcal{T}^P$ be the set of all candidate DTs obtained from T^P by selecting **one and only one** feature node for each of T^P's internal data nodes. We have $\mathcal{T}^P = \mathcal{T}^-$.*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y1_{(x,y)\in S_{n^d}}\big] + c_F\sqrt{P((x,y)\in S_{n^d})} < \lambda, \tag{5}$$

where $P((x,y)\in S_{n^d})$ and $\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y1_{(x,y)\in S_{n^d}}\big]$ can be further decomposed into Eq. 6 according to Bayes' theorem.

$$\begin{aligned} P((x,y)\in S_{n^d}) &= \prod_{n_{i,j}^{d'}\in e_{n^d}\backslash n_r} P(x_i = j|(x,y)\in S_{pa(pa(n_{i,j}^{d'}))}), \\ \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y1_{(x,y)\in S_{n^d}}\big] &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y|(x,y)\in S_{n^d}\big]P((x,y)\in S_{n^d}), \end{aligned} \tag{6}$$

where e_{n^d} is the node set containing all data nodes along the path from the FDT's root node to n^d, and $pa(pa(n_{i,j}^{d'}))$ indicates $n_{i,j}^{d'}$'s grandparent node.

Proof. The proof is elementary and is hence omitted. □

The proof of Theorem 5 can be finished by proving Lemmas 1-5 on the basis of Assumption 2-4. Lemmas 1 and 2 give the lower bound of visit time for each of FDT's nodes. Building on the results of Lemmas 1 and 2, Lemmas 3-5 show that for each of $\hat{T}_t^P$'s leaf node's parent feature node n^f, $\lim_{t\to\infty}\mathbb{E}[\hat{\sigma}_{n^f,t}] = L_{n^f}^*$ with high probablity.

Assumption 2. *$\forall i \in [l_T]$, $\mathbb{E}x_i^T$ is bounded by $0 < \alpha_b \leq \mathbb{E}x_i^T \leq \beta_b < 1$.*

Assumption 3. $y \in \{0, 1\}$.

Assumption 4. *For each of data node n^d in $\hat{T}_t$, we have*

$$|\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[y1_{(x,y)\in S_{n^d}}\big] + c_F\sqrt{P((x,y)\in S_{n^d})} - \lambda| \geq \epsilon_s,$$

where ϵ_s is a fixed postive real number.

Lemma 1. *Let n^f be $\hat{T}_t$'s feature node. Then we assumes that the i^{th} selection of n^f happens in the $t_{n^f,i}^{th}$ iteraions of MCTs-IS. With $\mathcal{G}_{t_{n^f,i},n^f} = \{$ n^f is selected in the $t_{n^f,i}^{th}$ iteration of MCTs-IS$\}$ and $1_{i,n_{\theta,\tau}^d} = 1(x_{t_{pa(n_{\theta,\tau}^d),i},p} = v|\mathcal{G}_{t_{pa(n_{\theta,\tau}^d),i},pa(n_{\theta,\tau}^d)})$, when $\delta_1 \in (0,1)$ and $N \geq \Omega(\log\frac{2K^2}{\delta_1})$, we have*

$$P(\sum_{i=1}^{N} 1_{i,n_{\theta,\tau}^d} \geq \frac{\alpha_b}{2}N) \geq 1 - \delta_1.$$

Proof. On the basis of the *Hoeffding's inequality*, we have

$$P(\sum_{i=1}^{N} 1_{i,n_{\theta,\tau}^d} \leq (\mathbb{E}1_{i,n_{\theta,\tau}^d} - \epsilon)N) \leq e^{-2\epsilon^2 N} \text{ and } P(\sum_{i=1}^{N} 1_{i,n_{\theta,\tau}^d} \geq (\mathbb{E}1_{i,n_{\theta,\tau}^d} + \epsilon)N) \leq e^{-2\epsilon^2 N} \tag{14}$$

First, we have

$$
\begin{aligned}
&P(for\ \forall\tau' \in [K_\theta], \sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau}} \geq \frac{\alpha_b}{2}N) \\
=&P(\sum_{\tau'\in[K_\theta]\backslash\{\tau\}} \sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} \leq (1-\frac{\alpha_b}{2})N \ \ \& \ \ for\ \forall\tau' \in [K_\theta]\backslash\{\tau\}, \sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \geq (\frac{\alpha_b}{2} - \mathbb{E}1_{i,n^d_{\theta,\tau'}})N) \\
=&P(\sum_{\tau'\in[K_\theta]\backslash\{\tau\}} \sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\sum_{v'\in[k_p]} \mathbb{E}1_{i,n^d_{\theta,\tau'}} \leq (1 - \sum_{\tau'\in[k_\theta]} \mathbb{E}1_{i,n^d_{\theta,\tau'}} - \frac{\alpha_b}{2})N \\
&\ \ \& \ \ for\ \forall\tau' \in [K_\theta]\backslash\{\tau\}, \sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \geq (\frac{\alpha_b}{2} - \mathbb{E}1_{i,n^d_{\theta,\tau'}})N) \\
\geq&P(for\ \forall\tau' \in [K_\theta]\backslash\{\tau\}, \frac{(\mathbb{E}1_{i,n^d_{\theta,\tau}} - \frac{\alpha_b}{2})N}{K_\theta - 1} \geq \sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \geq (\frac{\alpha_b}{2} - \mathbb{E}1_{i,n^d_{\theta,\tau'}})N) \\
\geq&P(for\ \forall\tau' \in [K_\theta]\backslash\{\tau\}, \frac{(\mathbb{E}1_{i,n^d_{\theta,\tau}} - \frac{\alpha_b}{2})N}{K_\theta - 1} \geq \sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \geq \frac{(\frac{\alpha_b}{2} - \mathbb{E}1_{i,n^d_{\theta,\tau'}})N}{K_\theta - 1}) \\
\geq&1 - \sum_{\tau'\in[K_\theta]\backslash\{\tau\}} P(\sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \leq -\frac{(\mathbb{E}1_{i,n^d_{\theta,\tau'}} - \frac{\alpha_b}{2})N}{K_\theta - 1}) \\
&- \sum_{\tau'\in[K_\theta]\backslash\{\tau\}} P(\sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \geq \frac{(\mathbb{E}1_{i,n^d_{\theta,\tau}} - \frac{\alpha_b}{2})N}{K_\theta - 1})
\end{aligned}
\tag{15}
$$

With $K = \max_{\theta\in[q]} K_\theta$, by combining Eqs .14 and 15, we have

$$
\begin{aligned}
&1 - \sum_{\tau'\in[K_\theta]\backslash\{\tau\}} P(\sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \leq -\frac{(\mathbb{E}1_{i,n^d_{\theta,\tau'}} - \epsilon_1)N}{K_\theta - 1}) \\
&- \sum_{\tau'\in[K_\theta]\backslash\{\tau\}} P(\sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \geq \frac{(\mathbb{E}1_{i,n^d_{\theta,\tau}} - \epsilon_1)N}{K_\theta - 1}) \\
\geq&1 - \sum_{\tau'\in[K_\theta]\backslash\{\tau\}} P(\sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \leq -\frac{\alpha_b N}{2K_\theta - 2}) - \sum_{\tau'\in[K_\theta]\backslash\{\tau\}} P(\sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau'}} - N\mathbb{E}1_{i,n^d_{\theta,\tau'}} \geq \frac{\alpha_b N}{2K_\theta - 2}) \\
\geq&1 - 2(K_\theta - 1)e^{\frac{-\alpha_b^2 N}{2(K_\theta-1)^2}} \\
\geq&1 - 2Ke^{\frac{-\alpha_b^2 N}{2K^2}}
\end{aligned}
$$

With $N \geq \Omega(\log \frac{2K^2}{\delta_1})$, we finally have $P(\sum_{i=1}^{N} 1_{i,n^d_{\theta,\tau}} \geq \frac{\alpha_b}{2}N) \geq 1 - \delta_1$. □

Lemma 2. *Define the event $\mathcal{A} = \{$The FDT $\hat{T}_t$ fitted by MCTs-IS has already been fully expanded, and for each of $\hat{T}_t$'s data node n^d, $s_{n^d} < 1$ and $s_{n^d} > 1$ are correctly estimated.$\}$. When iteration number is greater than $t_{\mathcal{A}}$, $\mathcal{A}$ happens with high probability.*

Proof. Based on Theorem 4 in (Kocsis et al., 2006), for each feature node n^f in $\hat{T}_t$, we have $V_{n^f,t} = \Omega(\log V_{pa(n^f),t})$, while for each data node n^d in $\hat{T}_t$, according to Lemma 1, if $V_{n^d,t}$ is large enough, we have $P(for\ \forall\tau' \in [K_\theta], V_{n^d_{\theta,\tau'},t} \geq \frac{\alpha_b}{2} V_{pa(n^d_{\theta,\tau'}),t}) \geq 1 - \delta$, where δ is a small positive real value. Therefore, if the number of iterations of MCTs-IS is large enough, with large probability, the FDT tree $\hat{T}_t$ can be fully expanded, and under Assumption 4, for each of $\hat{T}_t$'s data node n^d, $s_{n^d} < 1$ and $s_{n^d} > 1$ can be correctly estimated according to the *hoeffding inequality*. □

Lemma 3. *Let $\hat{T}_t^P$ be the FDT after pruning all $\hat{T}_t$'s feature nodes n^f with $n^f \notin l(pa(n^f))$, and n^f be $\hat{T}_t^P$'s leaf node's father node conditioned on $\mathcal{A}$. With $\mathcal{G}_{t_{n^f,i},n^f} = \{$ n^f is selected in the $t^{th}_{n^f,i}$ iteration of MCTs-IS$\}$ and $1_{i,n^d_{\theta,\tau}} = 1(x_{t_{pa(n^d_{\theta,\tau}),i},p} = v|\mathcal{G}_{t_{pa(n^d_{\theta,\tau}),i},pa(n^d_{\theta,\tau})})$, we have*

$$|\frac{1}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\mathbb{E}[\sum_{n^d_{\theta,\tau}\in c(n^f)}1_{i,n^d_{\theta,\tau}}(y_{t_{n^f,i}}-\frac{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}y_{t_{n^f,i'}}}{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}})^2]-L^*_{n^f}| \leq \mathcal{O}(\frac{\log V_{n^f,t_{max}}}{V_{n^f,t_{max}}}). \tag{16}$$

Proof. With $\mu_{n^d_{\theta,\tau}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y|(x,y)\in S_{n^d_{\theta,\tau}}]$, the left hand side of Eq. 16 can be expanded to

$$\begin{aligned}
&\frac{1}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\mathbb{E}[\sum_{n^d_{\theta,\tau}\in c(n^f)}1_{i,n^d_{\theta,\tau}}(y_{t_{n^f,i}}-\frac{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}y_{t_{n^f,i'}}}{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}})^2|\mathcal{A}]\\
=&\mathbb{E}[\frac{1}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\sum_{n^d_{\theta,\tau}\in c(n^f)}1_{i,n^d_{\theta,\tau}}(y_{t_{n^f,i}}-\mu_{n^d_{\theta,\tau}})^2|\mathcal{A}]\\
&+\mathbb{E}[\frac{1}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\sum_{n^d_{\theta,\tau}\in c(n^f)}1_{i,n^d_{\theta,\tau}}(\mu_{n^d_{\theta,\tau}}-\frac{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}y_{t_{n^f,i'}}}{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}})^2|\mathcal{A}]\\
&+\mathbb{E}[\frac{2}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\sum_{n^d_{\theta,\tau}\in c(n^f)}1_{i,n^d_{\theta,\tau}}(y_{t_{n^f,i}}-\mu_{n^d_{\theta,\tau}})(\mu_{n^d_{\theta,\tau}}-\frac{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}y_{t_{n^f,i'}}}{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}})|\mathcal{A}]
\end{aligned} \tag{17}$$

The first term of Eq. 17 is $L^*_{n^f}$, while the second term can be further expanded as

$$\begin{aligned}
&\frac{1}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\sum_{n^d_{\theta,\tau}\in c(n^f)}\mathbb{E}[1_{i,n^d_{\theta,\tau}}(\frac{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}(\mu_{n^d_{\theta,\tau}}-y_{t_{n^f,i'}})}{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}})^2|i,\theta,\tau,\mathcal{A}]\\
=&\frac{1}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\sum_{n^d_{\theta,\tau}\in c(n^f)}\mathbb{E}[1_{i,n^d_{\theta,\tau}}(\frac{\sum_{i'\in[i]\setminus i''}1_{i',n^d_{\theta,\tau}}(\mu_{n^d_{\theta,\tau}}-y_{t_{n^f,i'}})}{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}})^2|i'',i,\theta,\tau,\mathcal{A}]\\
&+\frac{1}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\sum_{n^d_{\theta,\tau}\in c(n^f)}\mathbb{E}[1_{i,n^d_{\theta,\tau}}\frac{(1_{i'',n^d_{\theta,\tau}}(\mu_{n^d_{\theta,\tau}}-y_{t_{n^f,i''}}))^2}{(\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}})^2}|i'',i,\theta,\tau,\mathcal{A}]\\
&+\frac{2}{V_{n^f,t_{max}}}\sum_{i=1}^{V_{n^f,t_{max}}}\sum_{n^d_{\theta,\tau}\in c(n^f)}\mathbb{E}[1_{i,n^d_{\theta,\tau}}(\frac{(1_{i'',n^d_{\theta,\tau}}(\mu_{n^d_{\theta,\tau}}-y_{t_{n^f,i''}}))\sum_{i'\in[i]\setminus i''}1_{i',n^d_{\theta,\tau}}(\mu_{n^d_{\theta,\tau}}-y_{t_{n^f,i'}})}{\sum_{i'=1}^{i}1_{i',n^d_{\theta,\tau}}})|i'',i,\theta,\tau,\mathcal{A}]
\end{aligned} \tag{18}$$

The third term of Eq. 18 equals 0 because $y_{t_{n^f,i''}}$ is independent with $y_{t_{n^f,i'}}$. After fully expanding the first term of Eq. 18 recursively, we get

$$
\begin{aligned}
&\frac{1}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} \mathbb{E}[1_{i,n^d_{\theta,\tau}} (\mu_{n^d_{\theta,\tau}} - \frac{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}} y_{t_{n^f,i'}}}{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}}})^2 | i, \theta, \tau, \mathcal{A}] \\
=&\frac{1}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} \mathbb{E}[1_{i,n^d_{\theta,\tau}} \frac{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}} (\mu_{n^d_{\theta,\tau}} - y_{t_{n^f,i'}})^2}{(\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}})^2} | i, \theta, \tau, \mathcal{A}] \\
\leq&\frac{1}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} \mathbb{E}[\frac{1_{i,n^d_{\theta,\tau}}}{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}}} | i, \theta, \tau, \mathcal{A}] \\
=&\frac{1}{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} \sum_{i=1}^{V_{n^f,t_{max}}} P(1_{i,n^d_{\theta,\tau}} = 1 | i, \theta, \tau, \mathcal{A}) \mathbb{E}[\frac{1}{1 + \sum_{i' \in [i-1]} 1_{i',n^d_{\theta,\tau}}} | 1_{i,n^d_{\theta,\tau}} = 1, i, \theta, \tau, \mathcal{A}]
\end{aligned}
\tag{19}
$$

Since $1_{i,n^d_{p,v}}$ is a bernoulli random variable, we have

$$
\frac{1}{i} \leq \mathbb{E}[\frac{1}{1 + \sum_{i' \in [i-1]} 1_{i',n^d_{\theta,\tau}}} | 1_{i,n^d_{\theta,\tau}} = 1, i, \theta, \tau, \mathcal{A}] = \frac{1 - (1 - \mathbb{E}[1_{i,n^d_{\theta,\tau}} | i, \theta, \tau, \mathcal{A}])^i}{i \mathbb{E}[1_{i,n^d_{\theta,\tau}} | i, \theta, \tau, \mathcal{A}]} \leq \frac{1}{i \alpha_b}
\tag{20}
$$

Then based on $\sum_{i=1}^{N} \frac{1}{i} = \Theta(\log N)$. Eq. 19 can be further upper bounded by

$$
\begin{aligned}
&\frac{1}{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} \sum_{i=1}^{V_{n^f,t_{max}}} \mathbb{E}[\frac{1}{1 + \sum_{i' \in [i-1]} 1_{i',n^d_{\theta,\tau}}} | 1_{i,n^d_{\theta,\tau}} = 1, i, \theta, \tau, \mathcal{A}] \\
\leq&\frac{1}{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} \sum_{i=1}^{V_{n^f,t_{max}}} \frac{1}{i \alpha_b} \leq \mathcal{O}(\frac{\log V_{n^f,t_{max}}}{V_{n^f,t_{max}}})
\end{aligned}
$$

The last term of Eq. 17 can be upper-bounded by

$$
\begin{aligned}
&\mathbb{E}[\frac{2}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} 1_{i,n^d_{\theta,\tau}} (y_{t_{n^f,i}} - \mu_{n^d_{\theta,\tau}}) (\mu_{n^d_{\theta,\tau}} - \frac{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}} y_{t_{n^f,i'}}}{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}}}) | \mathcal{A}] \\
=&\mathbb{E}[\frac{2}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} 1_{i,n^d_{\theta,\tau}} (y_{t_{n^f,i}} - \mu_{n^d_{\theta,\tau}}) (\frac{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}} (\mu_{n^d_{\theta,\tau}} - y_{t_{n^f,i'}})}{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}}}) | \mathcal{G}_{i,n^f}, \mathcal{A}] \\
=& -\mathbb{E}[\frac{2}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} 1_{i,n^d_{\theta,\tau}} \frac{(y_{t_{n^f,i}} - \mu_{n^d_{\theta,\tau}})^2}{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}}} | \mathcal{G}_{i,n^f}, \mathcal{A}] \\
&+ \mathbb{E}[\frac{2}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} 1_{i,n^d_{\theta,\tau}} (y_{t_{n^f,i}} - \mu_{n^d_{\theta,\tau}}) (\frac{\sum_{i'=1}^{i-1} 1_{i',n^d_{\theta,\tau}} (\mu_{n^d_{\theta,\tau}} - y_{t_{n^f,i'}})}{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}}}) | \mathcal{G}_{i,n^f}, \mathcal{A}]
\end{aligned}
\tag{21}
$$

Eq. 21's second term of equals 0, and its first term is lower bounded by $-\Omega(\frac{\log V_{n^f,t_{max}}}{V_{n^f,t_{max}}})$. Since Eq. 19 is greater than 0 while Eq. 21 is smaller than 0, we have

$$
|\frac{1}{V_{n^f,t_{max}}} \sum_{i=1}^{V_{n^f,t_{max}}} \mathbb{E}[\sum_{n^d_{\theta,\tau} \in c(n^f)} 1_{i,n^d_{\theta,\tau}} (y_{t_{n^f,i}} - \frac{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}} y_{t_{n^f,i'}}}{\sum_{i'=1}^{i} 1_{i',n^d_{\theta,\tau}}})^2] - L^*_{n^f}| \leq \mathcal{O}(\frac{\log V_{n^f,t_{max}}}{V_{n^f,t_{max}}}).
$$

□

Lemma 4. *Let $X_1, ..., X_N$ be independent random variables on $\{0, 1\}$. With*

$$f(X_1, ..., X_N) = \sum_{n=1}^{N} (X_n - \sum_{i=1}^{n} \frac{X_i}{n})^2,$$

we have :

$$P(\sum_{n=1}^{N} (\frac{n-1}{n} X_n - \sum_{i=1}^{n-1} \frac{X_i}{n})^2 - \mathbb{E}\big[\sum_{n=1}^{N} (\frac{n-1}{n} X_n - \sum_{i=1}^{n-1} \frac{X_i}{n})^2\big]) \leq e^{-\frac{2t^2}{81N}}$$

Proof. First, function f can be written as

$$f(X_1, ..., X_N) = \sum_{n=1}^{N} (\frac{n-1}{n} X_n - \sum_{i=1}^{n-1} \frac{X_i}{n})^2.$$

Then for every $i = 1, ..., n$ and every $(x_1, ..., x_n)$,$(x'_1, ..., x'_n)$ that differ only in the i-th coordinate ($x_j = x'_j$ for all $j \neq i$), we have

$$\begin{aligned}
&|f(X_1, ..., X_N) - f(X'_1, ..., X'_N)| \\
=&|\sum_{n=i+1}^{N} (\frac{n-1}{n} X_n - \sum_{j \in [n-1] \setminus \{i\}} \frac{X_j}{n} - \frac{X_i}{n})^2 - (\frac{n-1}{n} X_n - \sum_{j \in [n-1] \setminus \{i\}} \frac{X_j}{n} - \frac{X'_i}{n})^2 \\
&+ (\frac{i-1}{i} X_i - \sum_{j=1}^{i-1} \frac{X_j}{i})^2 - (\frac{i-1}{i} X'_i - \sum_{j=1}^{i-1} \frac{X_j}{i})^2| \\
\leq&|\sum_{n=i+1}^{N} 2(\frac{n-1}{n} X_n - \sum_{j \in [n-1] \setminus \{i\}} \frac{X_j}{n})(\frac{X'_i - X_i}{n})| + |\sum_{n=i+1}^{N} (\frac{X_i^2 - X_i'^2}{n^2})| \\
&+ |2(\sum_{j=1}^{i-1} \frac{X_j}{i})(\frac{i-1}{i}(X'_i - X_i))| + |(\frac{i-1}{i})^2 (X_i^2 - X_i'^2)| \\
\leq&3(\frac{i-1}{i})^2 + 2 \sum_{n=i+1}^{N} \frac{n-1}{n^2} + \sum_{n=i+1}^{N} \frac{1}{n^2} \\
\leq&3 + 2 \sum_{n \in [N]} \frac{n-1}{n^2} + \sum_{n \in [N]} \frac{1}{n^2} \\
\leq&3 + 2\log(N+1) + \frac{\pi}{6}
\end{aligned}$$

For the sake of simplicity, we assume t that when $N \geq t_{\mathcal{A}}$, $3 + 2\log(N+1) + \frac{\pi}{6} \leq 3\log(N)$. Then according to McDiarmid's inequality, we have

$$\begin{aligned}
&P(\sum_{n=1}^{N} (\frac{n-1}{n} X_n - \sum_{i=1}^{n-1} \frac{X_i}{n})^2 - \mathbb{E}\big[\sum_{n=1}^{N} (\frac{n-1}{n} X_n - \sum_{i=1}^{n-1} \frac{X_i}{n})^2\big] \geq 3\log N \sqrt{N \log(\frac{1}{\delta_1})}) \leq \delta_1, \\
&P(\sum_{n=1}^{N} (\frac{n-1}{n} X_n - \sum_{i=1}^{n-1} \frac{X_i}{n})^2 - \mathbb{E}\big[\sum_{n=1}^{N} (\frac{n-1}{n} X_n - \sum_{i=1}^{n-1} \frac{X_i}{n})^2\big] \leq -3\log N \sqrt{N \log(\frac{1}{\delta_1})}) \leq \delta_1.
\end{aligned}$$

□

Lemma 5. *With*

$$\begin{aligned}
\hat{\sigma}'_{n^d, t_{max}} &= \frac{1}{V_{n^d, t_{max}}} \sum_{i=1}^{V_{n^d, t_{max}}} (y_{t_{n^d, i}} - \frac{\sum_{i'=1}^{i} 1_{i', n^d} y_{t_{n^d, i'}}}{\sum_{i'=1}^{i} 1_{i', n^d}})^2 \\
\hat{\sigma}'_{n^f, t_{max}} &= \frac{1}{V_{n^f, t_{max}}} \sum_{i=1}^{V_{n^f, t_{max}}} \sum_{n^d_{\theta,\tau} \in c(n^f)} 1_{i, n^d_{\theta,\tau}} (y_{t_{n^f, i}} - \frac{\sum_{i'=1}^{i} 1_{i', n^d_{\theta,\tau}} y_{t_{n^f, i'}}}{\sum_{i'=1}^{i} 1_{i', n^d_{\theta,\tau}}})^2
\end{aligned}$$

We have

$$P(|V_{n^f,t_{max}}\hat{\sigma}'_{n^f,t_{max}} - V_{n^f,t_{max}}\mathbb{E}[\hat{\sigma}'_{n^f,t_{max}}]| \geq 3K\log\epsilon_1 V_{n^f,t_{max}}\sqrt{\epsilon_1 V_{n^f,t_{max}}\log(\frac{1}{\delta})}) \leq 3\delta$$

Proof. According to Lemma 1, the event $\mathcal{C}_{V_{n^f,t_{max}}} = \{\ \forall n^d \in l(n^f), V_{n^d,t_{max}} \geq \epsilon_1 V_{n^f,t_{max}}\}$ happens with large probability when $V_{n^f,t_{max}}$ is large enough. Let $P(\mathcal{C}_{V_{n^f,t_{max}}}) \leq \delta_1$, with Lemma 4, we have

$$\begin{aligned}
&P(V_{n^f,t_{max}}\hat{\sigma}'_{n^f,t_{max}} - V_{n^f,t_{max}}\mathbb{E}[\hat{\sigma}'_{n^f,t_{max}}] \geq 3K\log\epsilon_1 V_{n^f,t_{max}}\sqrt{\epsilon_1 V_{n^f,t_{max}}\log(\frac{1}{\delta_1})}) \\
\leq& P(\mathcal{C}_{V_{n^f,t_{max}}})P(V_{n^f,t_{max}}\hat{\sigma}'_{n^f,t_{max}} - V_{n^f,t_{max}}\mathbb{E}[\hat{\sigma}'_{n^f,t_{max}}] \geq 3K\log\epsilon_1 V_{n^f,t_{max}}\sqrt{\epsilon_1 V_{n^f,t_{max}}\log(\frac{1}{\delta_1})}|\mathcal{C}_{V_{n^f,t_{max}}}) \\
&+ P(\overline{\mathcal{C}_{V_{n^f,t_{max}}}}) \\
\leq& \sum_{n^d\in l(n^f)} P(V_{n^d,t_{max}}\hat{\sigma}'_{n^d,t_{max}} - V_{n^d,t_{max}}\mathbb{E}[\hat{\sigma}'_{n^d,t_{max}}] \geq 3\log\epsilon_1 V_{n^d,t_{max}}\sqrt{\epsilon_1 V_{n^d,t_{max}}\log(\frac{1}{\delta_1})}|\mathcal{C}_{n^f,t_{max}}) + \delta_1 \\
\leq& (K+1)\delta_1
\end{aligned}$$

Similarly, we have

$$P(V_{n^f,t_{max}}\hat{\sigma}'_{n^f,t_{max}} - V_{n^f,t_{max}}\mathbb{E}[\hat{\sigma}'_{n^f,t_{max}}] \leq -3K\log\epsilon_1 V_{n^f,t_{max}}\sqrt{\epsilon_1 V_{n^f,t_{max}}\log(\frac{1}{\delta_1})}) \leq (K+1)\delta_1,$$

which finishes the proof. □

Theorem 5. *In the t^{th} iteration of MCTs-IS, let $\hat{T}_t^P$ be the FDT after pruning all $\hat{T}_t$'s feature nodes n^f with $n^f \notin l(pa(n^f))$. For each of $\hat{T}_t^P$'s leaf nodes' parent feature node $n^{f'}$, if $L_{n^{f'}}$ is estimated by $\hat{\sigma}_{n^{f'},t}$, and for each $\hat{T}_t^P$'s internal data node n^d, $L^*_{n^d}$ is calculated on the basis of the estimated $L_{n^{f'}}$, a DT $\hat{T}_t^*$ can be obtained from $\hat{T}_t^P$, and we have $\lim_{t\to\infty} P(\hat{T}_t^* = T^*) = 1$.*

Proof. According to Lemma 2, when the iteration number approaches to infinity, we first have $T^P = \hat{T}_t^P$. In this case, let n^f be $\hat{T}_t^P$'s leaf nodes' parent node. Based on Lemmas 3 and 5, with large enough t, $L_{n^{f'}}$ can be estimated accurately. Besides, with the help of *hoeffding inequality*, $\hat{p}_{n^d_{i,j}}$ can accurately estimate $P(x_i = j|(x,y) \in S_{n^d})$ in Eq. 7. Therefore, following Eq. 7, for any $\hat{T}_t^P$'s internal data node n^d, $L^*_{n^d} = \min_{n^f\in c(n^d)} L^*_{n^f}$ holds for a large probability when the number of iteration approaches to infinity. □

B Appendix B

B.1 TYGEM Preprocessing

19×19 Go is a game played on a board that is represented as a 19×19 grid. A position on the Go board can be filled with either a white or black stone by a white or black player, and thus can be regarded as a tenary feature ($\{empty, black, white\}$). In the TYGEM 9D vs. 9D dataset, we first extract game records from 2006 to 2016. For each record, we then exclude all 19×19 board configurations whose centers are not empty and only keep positions within the 7×7 region around the center of the board (which is called Tengen) from each remaining configuration. Note that the 7×7 region is much larger than the widely used 3×3 region (also called 3×3 pattern) in the training of AI GO players. In the event the white player plays the next move, we reverse all colors of stones within each of the remaining 7×7 regions to ensure that the black player plays the next. Following that, we assign labels for these regions based on whether the player places a stone in the center during the next move (marked as 1) or not (marked as 0). Our final step is to randomly eliminate 99% records with labels of 0, to achieve a balanced data set with $\frac{Positive\ Label}{Negative\ Label} \approx \frac{1}{3}$.

B.2 IMPLEMENTATION DETAILS OF MCTS-IS AND GBDT

Before running MCTs-IS and GBDT, the 48 features of TYGEM were randomly divided into eight groups for MCTs-IS to be able to select high-way interactions. In this way, TYGEM's features are reduced to 8 while the number of possible values for each feature was increased from 3 to 3^6.

In MCTs-IS, s_{n^d} is used for pruning trees in MCTs-IS. Let n_3 be the set containing all data nodes with depth 3 in the FDT. For any MCTs-IS' data node n^d except the root node, $\exists n_{i,j}^{d'} \in n_3$, the calculation of s_{n^d} requires factors $P(x_i = j|(x,y) \in S_{pa(pa(n_{i,j}^{d'}))})$ and $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[y|(x,y) \in S_{n_{i,j}^{d'}}\right]$ (see Eq. 6 for details). We estimated the two factors for each $n_{i,j}^{d'} \in n_3$ before running MCTs-IS based on the training set. During MCTs-IS, these estimations were fixed and can be directly applied to the calculation of s_{n^d}.

B.3 PARAMETER SETTING

The parameter settings of MCTs-IS and GBDT are given as follows.

1. **MCTs-IS**. N^T was set as 23 and 8 for the Avazu and TYGEM datasets, respectively. In the i^{th} FDT learning problem, all child feature nodes of the FDT's root node except the i^{th} one are pruned. Regarding MCTs-IS's hyper-parameters, we set C_p to 0.5, and υ and κ to 20 and 40, respectively. For c_F and λ, a grid search was conducted over $\{0.05, 0.005, 0.0005\}$ and $\{5e-3, 5e-4, 5e-5\}$ respectively.
2. **GBDT**. Consistent with MCTs-IS, 23 and 8 trees were built for Avazu and TYGEM datasets in GBDT, respectively. In our work, GBDT was implement by the package of Lightgbm, and Lightgbm limits the complexity of DTs according to the maximum number of leaves in the DT. Therefore a grid serach was conducted over $\{100, 500, 1000, 2000\}$ for the maximum number of leaf nodes.

LR, FM and FmFM were trained after the interaction selection with MCTs-IS and GBDT. For FmFM, we additionally use Mini-Batch Gradient Descent with Momentum (SGD with momentum) to to minimize the loss function, besides the Adam optimizer implemented in its open-source code. In SGD with momentum, 20 epochs were scheduled. The learning rate was set as 0.1, and was halved every five epochs . Additionally, we conduct a grid search on the penalty of the regularizer over $\{1e-5, 1e-6, 1e-7, 1e-8\}$.

B.4 MCTS-IS'S SELECTED INTERACTIONS IN TYGEM

All interactions with $s_{n^d} > \lambda$ (N=577) are collected and correlated to labels with χ^2 test. Afterwards, interactions significantly correlated with labels ($P < 0.05$, Bonferroni corrected) are selected and ranked according to the χ^2 statistic. Figure. 2 displays the top six interactions.

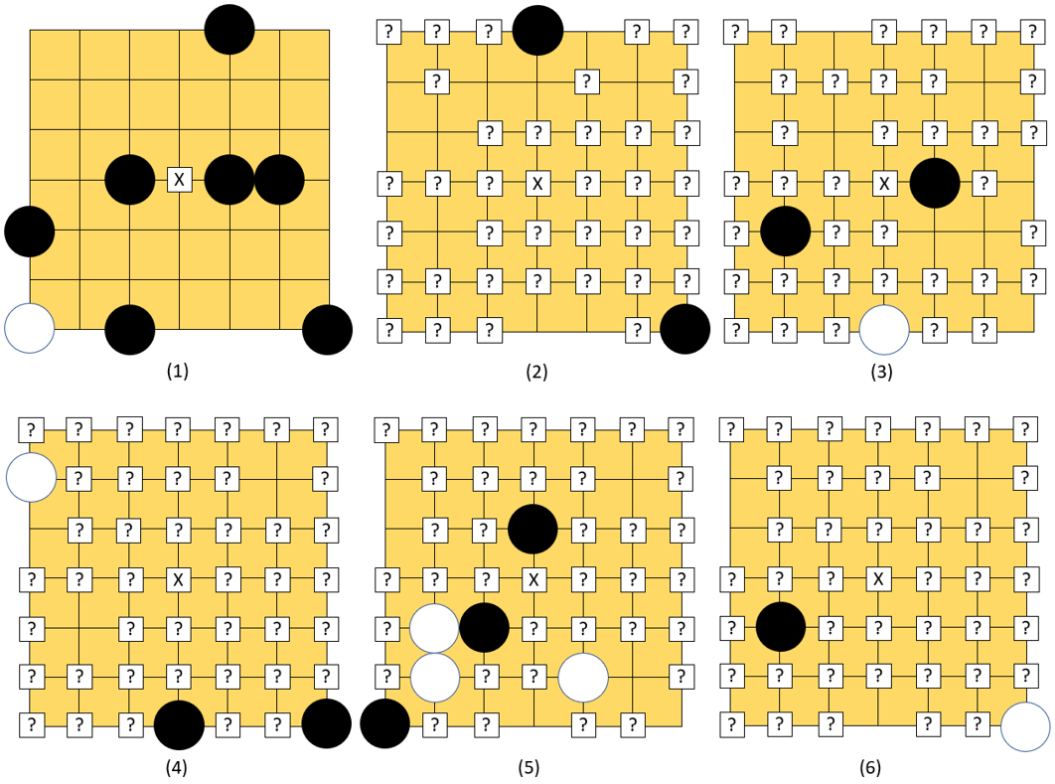

Figure 2: Interactions selected by MCTs-IS. Positions marked by "?" mean that these positions can be empty, or be placed by either white or black stones.