Fast Interpretable Greedy-Tree Sums

Yan Shuo Tan[a,1], **Chandan Singh**[b,g,1], **Keyan Nasseri**[b,1], **Abhineet Agarwal**[c,1], **James Duncan**[d], **Omer Ronen**[c], **Matthew Epland**[h], **Aaron Kornblith**[e,f], **and Bin Yu**[b,c,g,2]

[a]Department of Statistics and Data Science, National University of Singapore; [b]Department of Electrical Engineering and Computer Sciences, UC Berkeley; [c]Department of Statistics, UC Berkeley; [d]Graduate Group in Biostatistics, UC Berkeley; [e]Department of Emergency Medicine, UC San Francisco; [f]Department of Pediatrics, UC San Francisco; [g]Microsoft Research; [h]Overjet

This manuscript was compiled on July 7, 2023

Modern machine learning has achieved impressive prediction performance, but often sacrifices interpretability, a critical consideration in high-stakes domains such as medicine. In such settings, practitioners often use highly interpretable decision tree models, but these suffer from inductive bias against additive structure. To overcome this bias, we propose Fast Interpretable Greedy-Tree Sums (FIGS), which generalizes the CART algorithm to simultaneously grow a flexible number of trees in summation. By combining logical rules with addition, FIGS is able to adapt to additive structure while remaining highly interpretable. Extensive experiments on real-world datasets show that FIGS achieves state-of-the-art prediction performance. To demonstrate the usefulness of FIGS in high-stakes domains, we adapt FIGS to learn clinical decision instruments (CDIs), which are tools for guiding clinical decision-making. Specifically, we introduce a variant of FIGS known as G-FIGS that accounts for the heterogeneity in medical data. G-FIGS derives CDIs that reflect domain knowledge and enjoy improved specificity (by up to 20% over CART) without sacrificing sensitivity or interpretability. To provide further insight into FIGS, we prove that FIGS learns components of additive models, a property we refer to as disentanglement. Further, we show (under oracle conditions) that unconstrained tree-sum models leverage disentanglement to generalize more efficiently than single decision tree models when fitted to additive regression functions. Finally, to avoid overfitting with an unconstrained number of splits, we develop Bagging-FIGS, an ensemble version of FIGS that borrows the variance reduction techniques of random forests. Bagging-FIGS enjoys competitive performance with random forests and XGBoost on real-world datasets.

Interpretability | XGBoost | Clinical Decision Instrument | CART | Random Forest

1. Introduction

Modern machine learning methods such as random forests (1), gradient boosting (2, 3), and deep learning (4) display impressive predictive performance, but are complex and opaque, leading many to call them "black-box" models. Model interpretability is critical in many applications (5, 6), particularly in high-stakes settings such as clinical decision instrument (CDI) modeling. Interpretability allows models to be audited for general validation, errors, or biases, and therefore also more amenable to improvement by domain experts. Interpretability also facilitates counterfactual reasoning, which is the foundation of scientific insight, and it instills trust/distrust in a model when warranted. As an added benefit, interpretable models tend to be faster and more computationally efficient than black-box models.[*]

Decision trees are a prime example of interpretable models (2, 7–10). They can be easily visualized, memorized, and emulated by hand, even by non-experts, and thus fit naturally into high-stakes use-cases, such as decision-making in medicine (e.g. the emergency department[†]), law, and public policy. While decision trees have the potential to adapt to complex data, they are often outperformed by black-box models in terms of prediction performance. However, there is evidence that this performance gap is not intrinsic to interpretable models, e.g., see examples in (7, 10–12). In this paper, we identify an inductive bias of decision trees that causes its prediction performance to suffer in some instances, and design a new tree-based algorithm that overcomes this bias while preserving interpretability.

Our starting point is the observation that *decision trees can be statistically inefficient at fitting regression functions with additive components* (13). To illustrate this, consider the following toy example: $y = \mathbb{1}_{X_1>0} + \mathbb{1}_{X_2>0} \cdot \mathbb{1}_{X_3>0}$.[‡] The two components of this function can be individually implemented by trees with 1 split and 2 splits respectively. However, fitting their sum with a single tree requires at least 5 splits, as we are forced to make *duplicate subtrees*: a copy of the second tree has to be grown out of every leaf node of the first tree (see Fig 1). Indeed, given independent tree functions $f_1, \ldots, f_k$, in order for a single tree f to implement their sum, we would generally need

$$\#\text{leaves}(f) \geq \prod_{k=1}^{K} \#\text{leaves}(f_k).$$

This need to grow a deep tree to capture additive structure implies two statistical weaknesses of decision trees when fitting them to additive data-generating mechanisms. First, growing a deep tree

[†]For example, in the popular tool mdcalc, over 90% of available CDIs take the form of a decision tree.

[‡]This toy model is an instance of a Local Spiky and Sparse (LSS) model (14), which is grounded in real biological mechanisms whereby an outcome is driven by interactions of inputs (e.g. bio-molecules) which display thresholding behavior.

Significance Statement

Machine learning holds the potential to solve many real-world problems, but interpretability is a necessary prerequisite for practitioners in high-stakes domains such as medicine and law. Decision trees are mostly interpretable, but often do not have the best accuracy. Here, we propose a new type of tree-based model, called Fast Interpretable Greedy-tree Sums (FIGS), that aims to produce accurate models which are also interpretable. FIGS extends CART, an existing algorithm that builds a single decision tree, to instead build a sum of trees with a chosen cap on the total number of rules. This extension allows FIGS to capture additive structure in data, achieving better accuracy than CART while remaining interpretable across a variety of real-world datasets.

The authors declare that no conflicts of interest exist.

[1]Y.T.(Author One), C.S. (Author Two), K.N. (Author Three), A.A. (Author Four) contributed equally to this work.

[2]To whom correspondence should be addressed: binyu@berkeley.edu

[*]FIGS is integrated into the imodels package github.com/csinva/imodels (7) with an sklearn-compatible API. Experiments for reproducing the results here can be found at github.com/Yu-Group/imodels-experiments.

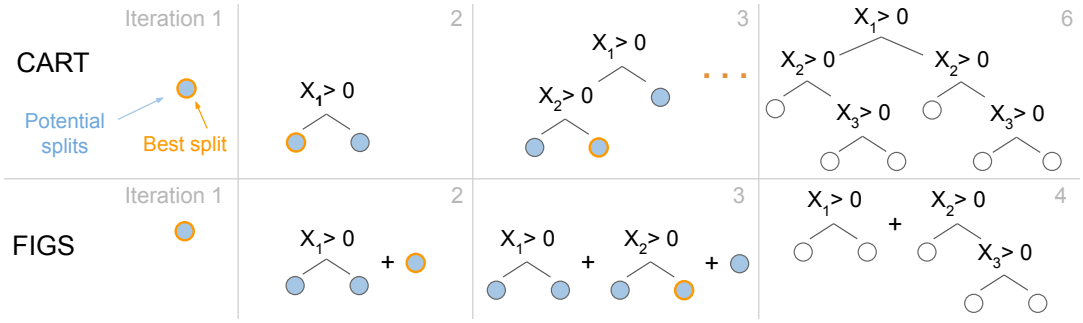


Fig. 1. FIGS algorithm overview for learning the toy function $y = \mathbb{1}_{X_1>0} + \mathbb{1}_{X_2>0} \cdot \mathbb{1}_{X_3>0}$. FIGS greedily adds one node at a time, considering splits not just in an individual tree but within a collection of trees in a sum. This can lead to much more compact models, as it reduces repeated splits (e.g., in the final CART model shown in the top-right).

greatly increases the probability of splitting on noisy features. Second, leaves in a deep tree contain fewer samples, implying that the predictions suffer from higher variance. These weaknesses could be mitigated if we fit a separate tree to each additive component of the generative mechanism and present the *tree-sum* as our fitted model. While existing ensemble methods such as random forests (1) and gradient boosting (2, 3) comprise tree-sums, they either fit each tree independently (random forests) or sequentially (gradient boosting), and are hence unable to disentangle the separate additive components.

To address these weaknesses of decision trees, we propose Fast Interpretable Greedy-Tree Sums (FIGS), a novel yet natural algorithm that is able to *grow a flexible number of trees simultaneously*. FIGS is based on a simple yet effective modification to Classification and Regression Trees (CART) (8), allowing it to adapt to additive structure (if present) by starting new trees, while still maintaining the ability of CART to adapt to higher-order interaction terms. By capping the total number of splits allowed, FIGS produces a model that is also easily visualized, memorized, and emulated by hand.

We performed extensive experiments across a wide array of real-world datasets to demonstrate the practical efficacy of FIGS. We compared FIGS with popular algorithms used to construct decision rule models (15). Taking the number of rules (splits in the case of trees) as a common measure of interpretability for this model class, we used the different algorithms to construct decision rule models of a prescribed level of interpretability, and found that FIGS often produced the model with the best prediction performance. An important use case for decision trees and other decision rule algorithms is the construction of CDIs. CDIs are models for predicting patient risk and are widely used by healthcare professionals to make rapid decisions in a clinical setting such as the emergency room. To apply FIGS to the medical domain, it is crucial that FIGS is able to adapt to heterogeneous data from diverse groups of patients. Tailoring models to various groups is often necessary since various groups of patients may differ dramatically and require distinct features for high predictive performance on the same outcome. While a naive solution is to fit a unique model to each group of patients, this is sample-inefficient and sacrifices valuable information that can be shared across groups. To mitigate this issue, we introduce a variant of FIGS, called Group Probability-Weighted Tree Sums, G-FIGS, which accounts for this heterogeneity while using all the samples available. Specifically, G-FIGS first fits a classifier (e.g., logistic regression) to predict group membership probabilities for each sample. Then, it uses these estimates as instance weights in FIGS to output a model for each group.

We used both FIGS and G-FIGS to construct CDIs for three pediatric emergency care datasets, and found that both have up to 10% higher specificity than CART while maintaining the same level of sensitivity. Further, G-FIGS improves performance over FIGS by over 3% for fixed levels of sensitivities. The features used in the fitted models agrees with medical domain knowledge. Moreover, G-FIGS learns a tree-sum model where each tree in the model represents a distinct clinical domain, providing a clear and organized framework for clinicians to use when assessing and treating patients. Next, we investigate the stability of G-FIGS to data perturbations. Stability to "reasonable" data perturbations is a key tenet of the Predictability, Computability, and Stability (PCS) framework (6, 16, 17), and a necessary prerequisite for the application of ML techniques in high-stakes domains. We show that G-FIGS learns a similar tree-sum model (i.e., a similar set of features) across data perturbations (e.g., introducing noise by randomly permuting labels).

Next, we investigate tree-sum models and FIGS theoretically. We prove generalization upper bounds for tree-sum models when given oracle access to the optimal tree structures. Whenever additive structure is present, this upper bound has a faster rate in the sample size n compared to the generalization lower bound for any decision tree proved in (13). Further, we establish in the large-sample limit that FIGS disentangles the additive components of the generative model, with each tree in the sum fitting a separate additive component.

Lastly, similarly to CART, FIGS overfits when allowed too many splits. Hence, we develop an ensemble version, called Bagging-FIGS, that borrows the bootstrap and feature subsetting strategies of random forests. We compare the prediction performance of Bagging-FIGS against random forests, XGBoost, and generalized additive models (GAMs) across a wide range of real-world datasets. Bagging-FIGS always maintains competitive performance with all other methods, further enjoying the best performance on several datasets.

In what follows, Sec 2 introduces FIGS and Sec 3 covers related work. Sec 4 contains our experimental results on real-world datasets showing that FIGS predicts well with very few splits. Sec 5 covers three CDI case studies using FIGS and G-FIGS. Sec 6 investigates the theoretical performance of tree-sum models and FIGS. Finally, Sec 7 introduces Bagging-FIGS and compares its prediction performance to other algorithms on real-world datasets.

2. FIGS: Algorithm description and run-time

Suppose we are given training data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. When growing a tree, CART chooses for each leaf node $\mathtt{t}$ the split s that maximizes the impurity decrease in the responses $\mathbf{y}$. For a given node

t, the impurity decrease has the expression

$$\hat{\Delta}(s, \mathtt{t}, \mathbf{y}) \coloneqq \sum_{\mathbf{x}_i \in \mathtt{t}} (y_i - \bar{y}_{\mathtt{t}})^2 - \sum_{\mathbf{x}_i \in \mathtt{t}_L} (y_i - \bar{y}_{\mathtt{t}_L})^2 - \sum_{\mathbf{x}_i \in \mathtt{t}_R} (y_i - \bar{y}_{\mathtt{t}_R})^2,$$

where $\mathtt{t}_L$ and $\mathtt{t}_R$ denote the left and right child nodes of $\mathtt{t}$ respectively, and $\bar{y}_{\mathtt{t}}, \bar{y}_{\mathtt{t}_L}, \bar{y}_{\mathtt{t}_R}$ denote the mean responses in each of the nodes. We call such a split s a *potential split*, and note that for each iteration of the algorithm, CART chooses the potential split with the largest impurity decrease.[§]

FIGS extends CART to greedily grow a tree-sum (see Algorithm 1). That is, at each iteration of FIGS, the algorithm chooses either to make a split on one of the current K trees $\hat{f}_1, \ldots, \hat{f}_K$ in the sum, or to add a new stump to the sum. To make this decision, it still applies the CART splitting criterion detailed above to identify a potential split in each leaf of each tree. However, to compute the impurity decrease for a given split, it substitutes the vector of *residuals* $r_i \coloneqq y - \sum_{k=1}^{K} \hat{f}_k(\mathbf{x}_i)$ for the vector of responses $\mathbf{y}$. FIGS makes only one split among the $K + 1$ trees: the one corresponding to the largest impurity decrease. The value of each of the new leaf nodes is then defined to be the mean residual for the samples it contains, added to the value of its parent node. If a new stump is created, the value at the root is defined to be zero. At inference time, we predict the response of an example by dropping it down each tree and summing the values of each leaf node containing it.

Algorithm 1 FIGS fitting algorithm.

```
FIGS(X: features, y: outcomes, stopping_threshold)
trees = []
while count_total_splits(trees) < stopping_threshold:
    all_trees = join(trees, build_new_tree()) # add new tree
    potential_splits = []
    for tree in all_trees:
        y_residuals = y – predict(all_trees)
        for leaf in tree:
            potential_split = split(X, y_residuals, leaf)
            potential_splits.append(potential_split)
    best_split = split_with_min_impurity(potential_splits)
    trees.insert(best_split)
```

Selecting the model's stopping threshold. Choosing a threshold on the total number of splits can be done similarly to CART: using a combination of the model's predictive performance (i.e., cross-validation) and domain knowledge on how interpretable the model needs to be. Alternatively, the threshold can be selected using an impurity decrease threshold (8) rather than a hard threshold on the number of splits. We discuss potential data-driven choices of the threshold in the Discussion (Sec 8).

Run-time analysis. The run-time complexity for FIGS to grow a model with m splits in total is $O(dm^2n^2)$, where d the number of features, and n the number of samples (see derivation in Appendix S1). In contrast, CART has a run-time of $O(dmn^2)$. Both of these worst-case run-times given above are quite fast, and the gap between them is relatively benign as we usually make a small number of splits to ensure interpretability.

[§] This corresponds to greedily minimizing the mean-squared-error criterion in regression and Gini impurity in classification.

Extensions. FIGS supports many natural modifications that are used in CART trees. For example, different impurity measures can be used; here we use Gini impurity for classification and mean-squared-error for regression. Additionally, FIGS could benefit from pruning, shrinkage (18), or by being used as part of an ensemble model (e.g., Bagging-FIGS). We discuss other extensions in Appendix S1.

3. Related work and its connections to FIGS

There is a long history of greedy methods for learning individual trees, e.g., C4.5 (9), CART (8), and ID3 (19). Recent work has proposed global optimization procedures rather than greedy algorithms for trees. These can improve performance given a fixed split budget but incur a high computational cost (20–22). However, due to the limitations of a single tree, all these methods have an inductive bias against additive structure (13, 23). Besides trees, there are a variety of other interpretable methods such as rule lists (24, 25), rule sets (26, 27), or generalized additive models (28); for an overview and Python implementation, see (7).

FIGS is related to backfitting (29), but differs crucially because it does not assume a fixed number of component features, nor does it require knowledge on which features are used by each component. Furthermore, FIGS does not update its component trees in a cyclic manner, but instead trees "compete for splits" at each iteration.

Similar to the work here are methods that learn an additive model of splits, where a split is defined to be an axis-aligned, rectangular region in the input space. RuleFit (30) is a popular method that learns a model by first extracting splits from multiple greedy decision trees fit to the data and then learning a linear model using those splits as features. FIGS is able to improve upon RuleFit by greedily selecting higher-order interactions when needed, rather than simply using all splits from some pre-specified tree depth. MARS (31) greedily learns an additive model of splines in a manner similar to FIGS, but loses some interpretability as a result of using splines rather than splits.

Also related to this work are tree ensembles, such as random forest (1), gradient-boosted trees (32), BART (33) and AddTree (34), all of which use ensembling as a way to boost predictive accuracy without focusing on finding an interpretable model. Loosely related are post-hoc methods which aim to help understand a black-box model (2, 35, 36), but these can display lack of fidelity to the original model, and also suffer from other problems (37).

4. FIGS results on real-world benchmark datasets

This section shows that FIGS enjoys strong prediction performance on several real-world benchmark datasets compared to popular algorithms for fitting decision rule models.

Benchmark datasets. For classification, we study four large datasets previously used to evaluate rule-based models (18, 38) along with the two largest UCI binary classification datasets used in Breiman's original paper introducing random forests (1, 39) (overview in Table 1). For regression, we study all datasets used in the random forest paper with at least 200 samples along with three of the largest non-redundant datasets from the PMLB benchmark (40). 80% of the data is used for training/3-fold cross-validation and 20% of the data is used for testing.

Baseline methods. For both classification and regression, **FIGS** is compared to CART, RuleFit, and Boosted Stumps (CART stumps learned via gradient-boosting). Furthermore, for classification we additionally compare against C4.5 and for regression we additionally compare against a CART model fitted using the mean-absolute-error

	Name	Samples	Features	Majority class
Classification	Readmission	101763	150	53.9%
	Credit (41)	30000	33	77.9%
	Recidivism	6172	20	51.6%
	Juvenile (42)	3640	286	86.6%
	German credit	1000	20	70.0%
	Diabetes (43)	768	8	65.1%
	Name	**Samples**	**Features**	**Mean**
Regression	Breast tumor (40)	116640	9	62.0
	CA housing (44)	20640	8	5.0
	Echo months (40)	17496	9	74.6
	Satellite image (40)	6435	36	7.0
	Abalone (45)	4177	8	29.0
	Diabetes (46)	442	10	346.0
	Friedman1 (31)	200	10	26.5
	Friedman2 (31)	200	4	1657.0
	Friedman3 (31)	200	4	1.6

Table 1. Real-world datasets analyzed here: classification (top panel), regression (bottom panel).

(MAE) splitting-criterion. We finally also add a black-box baseline comprising a Random Forest with 100 trees, which thus uses many more splits than all the other models.[¶]

FIGS predicts well with few splits. Fig 2 shows the models' performance results (on test data) as a function of the number of splits in the fitted model.[‖] We treat the number of splits as a common metric for the level of interpretability of each model. In practice, interpretability is context-dependent. However, the goal of this experiment is to sweep across a range of interpretability levels, and show the test performance had that level been selected.

The top two rows of Fig 2 show results for classification (measured using the area under the receiver operating characteristic (ROC) curve, i.e. AUC), and the bottom three rows show results for regression (measured using the coefficient of determination, denoted by R^2). On average, FIGS outperforms baseline models when the number of splits is very low. The performance gain from FIGS over other baselines is larger for the datasets with more samples (e.g., the top row of Fig 2), matching the intuition that FIGS performs better because of its increased flexibility. For two of the larger datasets (*Credit* and *Recidivism*), FIGS even outperforms the black-box Random Forest baseline, despite using less than 15 splits. For the smallest classification dataset (*Diabetes*), FIGS performs extremely well with very few (less than 10) splits but starts to overfit as more splits are added.

5. Learning CDIs via FIGS

A key domain problem involving interpretable models is the development of CDIs, which can assist clinicians in improving the accuracy, consistency, and efficiency of diagnostic strategies for sick and injured patients. Recent works have developed and validated CDIs using interpretable models, particularly in emergency medicine (47–50). As discussed earlier, applying machine learning models to the medical domain must account for the heterogeneity in the data that arises from the presence of diverse groups of patients (e.g., age groups, sex, treatment sites). Typically, this heterogeneity is dealt by fitting a separate model on each group. However, this strategy comes at the cost of losing samples, and discarding valuable information that can be shared amongst various groups. To mitigate this loss of power, we introduce a variant of FIGS, called group-probability weighted FIGS, or G-FIGS as follows.

G-FIGS: Fitting FIGS to multiple groups. As before, we assume a supervised learning setting with features X, outcome Y, and a group label G (e.g., treatment site or age group). G-FIGS is a two-step algorithm: (i) For a given group label g, G-FIGS first estimates group membership probabilities for each example (e.g., by fitting a logistic regression model to predict group-membership)[**] That is, it estimates $\mathbb{P}(G = g \mid X)$. (ii) For a given group g, G-FIGS then uses the group probabilities $\mathbb{P}(G = g \mid X)$ as sample weights when fitting FIGS to the *whole dataset*. This results in a group-specific model, but borrows information from samples in other groups. See Appendix S4 for more details, and a visual representation.

Datasets and data cleaning. Table 2 shows the CDI datasets under consideration here. They each constitute a large-scale multi-site data aggregation by the Pediatric Emergency Care Applied Research Network (PECARN), with a relevant clinical outcome (e.g., presence of traumatic brain injury). For each of these datasets, we group patients into two natural groups: patients with age <2 years and ≥ 2 years. This age-based threshold is commonly used for emergency-based diagnostic strategies (51), because it follows a natural stage of development, including a child's ability to participate in their care (e.g., ability to verbally communicate with their doctor). At the same time, the natural variability in early childhood development also creates opportunities to share information across this threshold. These datasets are non-standard for machine learning; as such, we spend considerable time cleaning, curating, and preprocessing these features along with medical expertise included in the authorship team.[††] We use 60% of the data for training, 20% for tuning hyperparameters (including estimation of each patient's group-membership $\mathbb{P}(\text{age} < 2 \text{ years} \mid X)$), and 20% for evaluating test performance.

Prediction metrics. Prediction performance is measured by comparing the specificity of a model when sensitivity is constrained to be above a given threshold, chosen to be $\{92\%, 94\%, 96\%, 98\%\}$. We opt for this metric because high levels of sensitivity are crucial for CDIs so as to avoid potentially life threatening false negatives (i.e., missing a diagnosis). For a given threshold, we would like to maximize specificity as false positives lead to unncessary resource utilization and can needlessly expose patients to the harmful effects of medical procedures such as radiation from a computed tomography (CT) scan.

Baseline methods. We compare FIGS and G-FIGS to two baselines: CART (8) and Tree-Alternating Optimization TAO (52)). For each baseline, we either (i) fit one model to all the training data or (ii) fit a separate model to each group (denoted with -SEP) – one to the patients with age <2 years and one for the patients with age ≥ 2 years. Additionally, for CART, we also fit a model in the style of G-FIGS, denoted as G-CART. Limits on the total number of splits for each model are varied over a range which yields interpretable models, from 2 to 16 maximum splits[‡‡] (full details of this and selection of other hyperparameters are inAppendix S5).

[¶] We also compare against Gradient-boosting with decision trees of depth 2, but find that it is outperformed by CART in this limited-split regime, so we omit these results for clarity. We also attempt to compare to optimal tree methods, such as GOSDT (20), but find that they are unable to fit the dataset sizes here.

[‖] For RuleFit, each term in the linear model is counted as one split.

[**] This is methodologically analogous to a propensity score in the causal inference literature.

[††] Details, along with the openly released clean data can be found in Appendix S5.

[‡‡] The choice of 16 splits is somewhat arbitrary, but we find that amongst 643 popular CDIs on mdcalc, 93% contain no more than 16 splits and 95% contain no more than 20 splits.

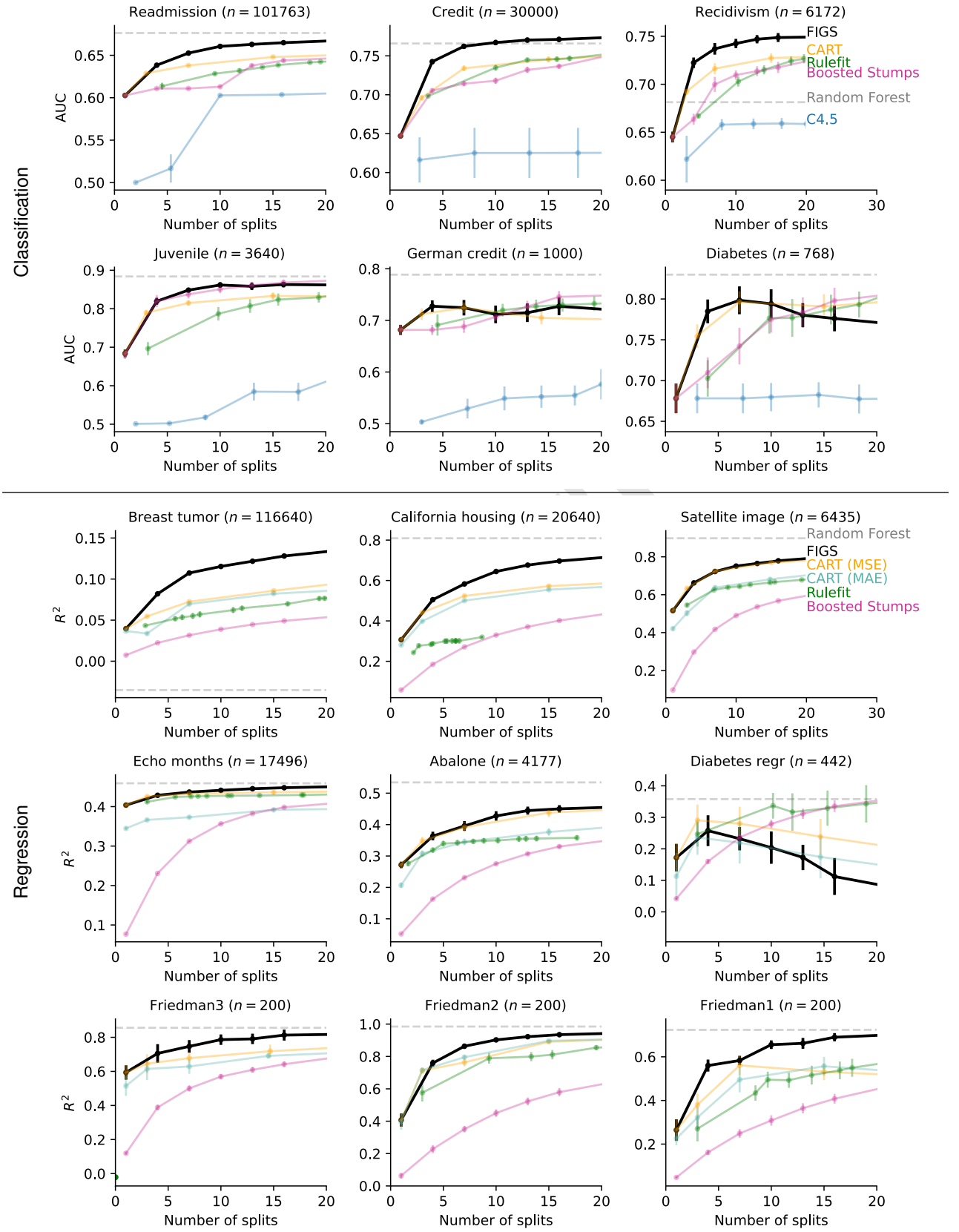


Fig. 2. FIGS performs extremely well on the test-set using very few splits, particularly when the dataset is large. Top two rows show results for classification datasets (measured by AUC of the ROC curve) and the bottom three rows show results for regression datasets (measured by R^2). Errors bars show standard error of the mean, computed over 6 random data splits.

Name	Patients	Features	Outcome (count)	Outcome (%)
TBI	42428	61	376	0.9%
IAI	12044	21	203	1.7%
CSI	3313	34	540	16.3%

Table 2. Clinical decision datasets for traumatic brain injury (TBI) (51), intra-abdominal injury (IAI) (50), and cervical spine injury (CSI) (53).

FIGS and G-FIGS predict well. Table 3 shows the prediction performance of FIGS, G-FIGS , and baseline methods. Further, we report the prediction performance of all methods for each age group separately (i.e, age <2 years and age ≥ 2 years) in Table S2 and Table S3 respectively. For high levels of sensitivity, G-FIGS generally improves the model's specificity against the baselines. Further, G-FIGS also improves specificity for each age group, often outperforming both the models that fit all the data as well as the model that fits data for each group separately. This suggests that using a different model for each group while sharing information across groups can lead to better prediction in a heterogeneous population.

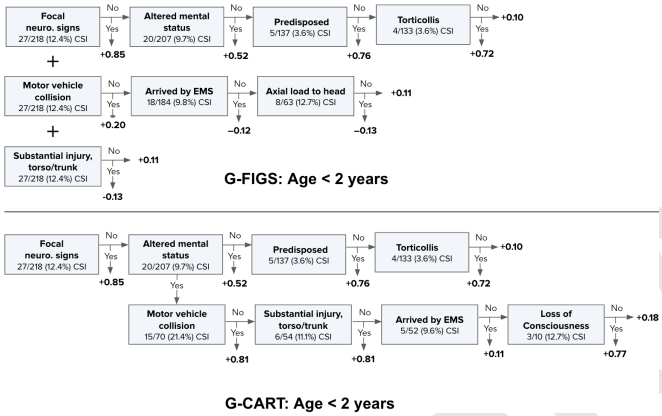


Fig. 3. Visualizes the G-FIGS and G-CART models fitted to CSI dataset for the age ≤ 2 years group. Both G-FIGS and G-CART use features that match medical knowledge. However, unlike G-CART, G-FIGS disentangles risk factors, with each tree representing a distinct clinical domain.

Features used by G-FIGS match medical knowledge. Fig 3 shows the G-FIGS model on the cervical spinal injury (CSI) dataset for patients with age <2 years, while Appendix S5 shows G-FIGS models for patients with age ≥ 2 years, and the other two datasets. The features used by the learned model match medical domain knowledge and partially agree with previous work (53); e.g., features such as *focal neurologic signs*, *altered mental status*, and *torticollis* are all known to increase the risk of CSI. Further, features unique to each group largely relate to the age cutoff; the <2 years age group features include those that clinicians can assess without asking the patient (e.g., *substantial torso injury*), while that of the ≥ 2 years age group features (visualized in Fig S3) require verbal responses (*neck pain*, *head pain*). This matches the medical intuition that non-verbal features should be more reliable features in the <2 years age group.

G-FIGS disentangles clinical risk factors. Fig 3 visualizes the G-FIGS and G-CART model for the <2 years age group on the CSI dataset. Both G-FIGS and G-CART utilize almost all the same features (apart from *Axial load to head*). However, unlike G-CART, G-FIGS disentangles risk factors with each tree representing a clinical domain: the top tree identifies signs and symptoms, the middle tree corresponds to the mode of injury and how the patient arrived at the hospital, and the bottom tree assesses the overall severity of the patient's injury patterns and any associated injuries that may be present. By disentangling clinical domains, the fitted G-FIGS model not only has stronger prediction performance than a single-tree model (see Table 2), but also provides a more medically intuitive framework for clinicians to use when assessing and treating patients. We provide another example of the ability of FIGS to disentangle risk factors on a diabetes classification dataset in Appendix S6. We note that there is no a priori reason that tree-sum models are more reflective of the true data generating process than single tree models. Instead, trust in the veracity of such a model should be based on good prediction performance as well as coherence with domain knowledge. In more detail, we recommend that practitioners follow the predictive, descriptive, and relevance (PDR) framework for investigating real-world scientific problems using interpretable machine learning models (6).

Stability Analysis of Learnt CDI. As discussed earlier, the PCS framework argues that stability to "reasonable" data perturbations is a key prerequisite for interpretability and deployment of machine learning methods in high-stakes domains. Here, we investigate the stability of G-FIGS on the CSI dataset. Specifically, we introduce noise by randomly swapping a percentage p of labels $\mathbf{y}$. We vary p between $\{1\%, 2.5\%, 5\%\}$. For each value of p, we measure stability by comparing the similarity of the features selected in the model trained on the perturbed data to the model displayed in Fig 3. In particular, the similarity of features is measured via the Jaccard distance. Further details of our experiments, and our results can be found in Appendix S5. Our results show that G-FIGS learns a similar model for each age group even for high values of p, indicating its stability.

6. Theoretical Investigations

We perform theoretical investigations to better understand the properties of FIGS and tree-sum models. Specifically, we show (under some oracle conditions) that if the regression function has an additive decomposition, then tree-sum models and FIGS are able to achieve optimal generalization upper bounds and disentangle additive components. In this section, we summarize these theoretical results, and defer the formal statements and proof details to Appendix S6.

Oracle generalization upper bounds. As discussed, *all* single-tree models have a squared error generalization error lower bound of $\Omega(n^{-2/(d+2)})$ when fitted to smooth additive models (13). To demonstrate the utility of tree-sum models in capturing additive structure, we provide generalization upper bounds of tree-sum models when their structure is chosen by an oracle. Specifically, we consider the typical supervised learning set-up $y = f(\mathbf{x}) + \epsilon$, where $\mathbf{x}$ is a random variable on $[0,1]^d$ and $\mathbb{E}\{\epsilon \mid \mathbf{x}\} = 0$. We assume $f(\mathbf{x}) = \sum_{k=1}^{K} f_k(\mathbf{x}_{I_k})$, where $I_1 \dots I_K$ are disjoint blocks of features, and $\mathbf{x}_{I_k}$ denotes the sub-vector of $\mathbf{x}$ comprising coordinates in I_k. Under this set-up, we show that if each component function f_k is smooth, and blocks of features are independent (i.e., $\mathbf{x}_{I_j} \perp\!\!\!\perp \mathbf{x}_{I_k}$ for $j \neq k$), then there exists a tree-sum model such that its squared error generalization upper bound scales as $O(Kn^{-2/(d_{max}+2)})$ where $d_{max} = max_k|I_k|$. It is instructive to consider two extreme cases: If $|I_k| = 1$ for each k, the upper bound scales as $O\left(dn^{-2/3}\right)$. On the other hand if $K = 1$, we have an upper bound of $O\left(n^{-2/(d+2)}\right)$. Both bounds match the well-known minimax rates for their respective inference problems (54). See Theorem 2 for the formal statement.

FIGS performs disentanglement. One potential reason for the success of FIGS is its ability to disentangle additive structure. We show

	Traumatic brain injury				Cervical spine injury				Intra-abdominal injury			
Sensitivity level:	92%	94%	96%	98%	92%	94%	96%	98%	92%	94%	96%	98%
TAO	6.2	6.2	0.4	0.4	41.5	21.2	0.2	0.2	0.2	0.2	0.0	0.0
TAO-SEP	26.7	13.9	10.4	2.4	32.5	7.0	5.4	2.5	12.1	8.5	2.0	0.0
CART	20.9	14.8	7.8	2.1	38.6	13.7	1.5	1.1	11.8	2.7	1.6	1.4
CART-SEP	26.6	13.8	10.3	2.4	32.1	7.8	5.4	2.5	11.0	9.3	2.8	0.0
G-CART	15.5	13.5	6.4	3.0	38.5	15.2	4.9	3.9	11.7	10.1	3.8	0.7
FIGS	23.8	18.2	12.1	0.4	39.1	33.8	24.2	**16.7**	**32.1**	13.7	1.4	0.0
FIGS-SEP	39.9	19.7	**17.5**	2.6	38.7	33.1	20.1	3.9	18.8	9.2	2.6	0.9
G-FIGS	**42.0**	**23.0**	14.7	**6.4**	**42.2**	**36.2**	**28.4**	15.7	29.7	**18.8**	**11.7**	**3.0**

Table 3. Best test-set specificity when sensitivity is constrained to be above a given level. G-FIGS provides the best performance overall in the high-sensitivity regime. -SEP models fit a separate model to each group, and generally outperform fitting a model to the entire dataset. G-CART follows the same approach as G-FIGS but uses weighted CART instead of FIGS for each final group model. Averaged over 10 random data splits into training, validation, and test sets, with hyperparameters chosen independently for each split.

that under the generative model discussed above, if FIGS splits nodes using population quantities (i.e., in the large-sample limit), then the set of features split upon in each fitted tree $\hat{f}_k$ is contained within I_k. The precise theorem statement be found in Theorem 1 in Appendix Appendix S6. By disentangling additive components, FIGS is able to avoid duplicate subtrees, leading to a more parsimonious model with better performance. We provide (partial) empirical justification of this in real-world datasets, by showing that FIGS helps to reduce the number of possibly redundant and repeated splits often observed in CART models (see Appendix S2.) As discussed earlier, we emphasize that disentanglement does not necessarily reflect the underlying data generating process. We refer the reader to Sec 5 for a larger discussion regarding interpreting disentanglement.

7. Bagging-FIGS

Growing deeper decision trees reduces bias but increases variance. Allowing more splits in a FIGS model has the same effect. In order to reduce variance, random forests averages the predictions from an ensemble of decision trees that are each grown in a slightly different manner (1). Bagging-FIGS averages the predictions from an ensemble of FIGS models, and makes use of the same variance reduction strategies as random forests: (i) Each FIGS model fit on a bootstrap resampled dataset. (ii) At each iteration of each FIGS model, a random subset of the original features is chosen, and the algorithm chooses the next split only from this subset of features. The effect of both these strategies have been studied empirically and theoretically (55–58).

Baseline methods and settings. In the remainder of this section, we will compare the prediction performance of FIGS and Bagging-FIGS against that of four other algorithms: random forest, XGBoost, and penalized iteratively reweighted least squares (PIRLS) on the log-likelihood of a generative additive model. All algorithms are fit using default settings[§§]. The number of features subsetted is a tuning parameter for random forest and Bagging-FIGS. For both algorithms, we set this to be $d/3$ for regression datasets and $\sqrt{d}$ for classification datasets. These are the default choices for RF. We fit Bagging-FIGS with 100 FIGS estimators.

Bagging-FIGS performs comparably to Random Forest and XGBoost Fig 4 shows the generalization performance of all the methods on the various real-world datasets introduced in Sec 4 and Sec 5. It shows that the performance of FIGS, measured by AUC can be improved via bagging and feature subsetting (Bagging-FIGS). In addition, Bagging-FIGS achieves comparable AUC to both XGBoost and random forest (on average improving over XGBoost by 0.015 and random forest by 0.013). For some datasets (e.g. *IAI pecarn*, *Recidivism*), Bagging-FIGS outperforms both baselines.

[§§] We use the implementation of PIRLS in `pygam` (59), with 20 splines term for each feature.

8. Discussion

FIGS is a powerful and natural extension to CART which achieves improved predictive performance over popular baseline tree-based methods across a wide array of datasets while maintaining interpretability by using very few splits. Furthermore, when the number of splits is unconstrained, an ensemble version of FIGS has prediction performance that compares favorably to random forest and XGBoost.

As a case study, we have shown how FIGS and G-FIGS can make an important step towards interpretable modeling of heterogeneous data in the context of high-stakes clinical decision-making. The fitted CDI models here show promise, but require external clinical validation before potential use. While our current work only explores age-based grouping, the behavior of G-FIGS with temporal, geographical, or demographic splits could be studied as well. Additionally, there are many methodological extensions to explore, such as data-driven identification of input data groups and schemes for feature weighting in addition to instance weighting. FIGS has many other natural extensions, some of which we detail below.

Optimization. Instead of using a greedy approach, one may perform global optimization algorithm over the class of tree-sum models. FIGS could include other moves such as pruning in addition to just adding more splits. This could also be embedded in a Bayesian framework, similar to Bayesian Additive Regression Trees (33).

Regularization. Using cross-validation (CV) to select the number of splits tends to select larger models. Future work can use criteria related to BIC (60) or stability in combination with CV (61) for selecting this threshold based on data. In future work, one could also vary the total number of splits and number of trees separately, helping to build prior knowledge into the fitting process. FIGS could be penalized via novel regularization techniques, such as regularizing individual leaves or regularizing a linear model formed from the splits extracted by FIGS (18).

Learning interactions. Interactions are known to be prevalent in biology and other fields, and therefore of key scientific interest (62). There has been recent interest in using random forests to learn interactions (14, 63, 64). However, since single decision trees are unable to disentangle additive structure, using random forests for interaction discovery might lead to a high false discovery rate. As a result,

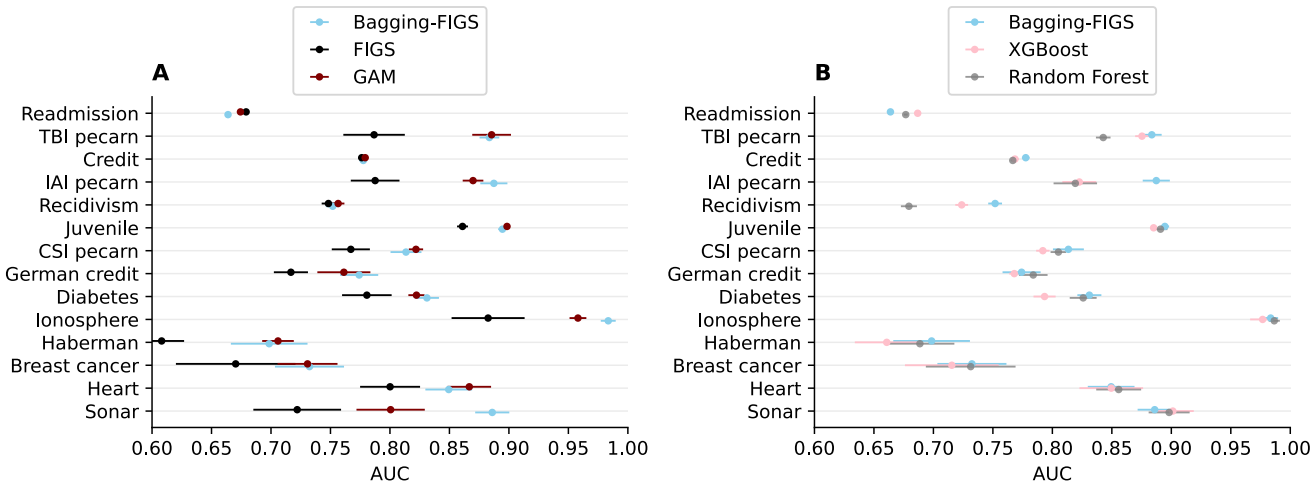


Fig. 4. Generalization performance across various real-world datasets with no constraints on the number of splits. **(A)** The performance of FIGS can usually be improved by fitting an ensemble of FIGS models using bagging and feature subsetting (Bagging-FIGS). The performance of Bagging-FIGS is comparable that of a GAM. **(B)** Bagging-FIGS achieves comparable performance to both XGBoost and random forest. Error bars show standard error of the mean, computed over 5 random data splits.

using FIGS in lieu of single decision trees might lead to improved interaction discovery. We leave this investigation to future work.

Model class. The class of FIGS models could be further extended to include linear terms or allow for summations of trees to be present at split nodes, rather than just at the root.

We hope FIGS and G-FIGS can pave the way towards more transparent and interpretable modeling that can improve machine-learning practice, particularly in high-stakes domains such as medicine, law, and policy making.

9. Acknowledgements

We gratefully acknowledge partial support from NSF TRIPODS Grant 1740855, DMS-1613002, 1953191, 2015341, 2209975, IIS 1741340, ONR grant N00014-17-1-2176, the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370, NSF grant 2023505 on Collaborative Research: Foundations of Data Science Institute (FODSI), the NSF and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning through awards DMS-2031883 and 814639, and a Weill Neurohub grant. YT was partially supported by NUS Start-up Grant A-8000448-00-00. AK was supported by the Eunice Kennedy Shriver National Institute of Child Health and Human Development of the National Institutes of Health under Award Number K23HD110716–01.

1. Breiman L (2001) Random forests. *Machine learning* 45(1):5–32.
2. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics* pp. 1189–1232.
3. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. pp. 785–794.
4. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *nature* 521(7553):436–444.
5. Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5):206–215.
6. Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B (2019) Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* 116(44):22071–22080.
7. Singh C, Nasseri K, Tan YS, Tang T, Yu B (2021) imodels: a python package for fitting interpretable models. *Journal of Open Source Software* 6(61):3192.
8. Breiman L, Friedman J, Olshen R, Stone CJ (1984) *Classification and regression trees*. (Chapman and Hall/CRC).
9. Quinlan JR (2014) *C4. 5: programs for machine learning*. (Elsevier).
10. Rudin C, et al. (2021) Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251*.
11. Ha W, Singh C, Lanusse F, Upadhyayula S, Yu B (2021) Adaptive wavelet distillation from neural networks through interpretations. *Advances in Neural Information Processing Systems* 34.
12. Mignan A, Broccardo M (2019) One neuron versus deep learning in aftershock prediction. *Nature* 574(7776):E1–E3.
13. Tan YS, Agarwal A, Yu B (2021) A cautionary tale on fitting decision trees to data from additive models: generalization lower bounds. *arXiv preprint arXiv:2110.09626*.
14. Behr M, Wang Y, Li X, Yu B (2021) Provable boolean interaction recovery from tree ensemble obtained via random forests. *arXiv preprint arXiv:2102.11800*.
15. Molnar C (2020) *Interpretable machine learning*. (Lulu. com).
16. Yu B (2013) Stability. *Bernoulli* 19(4):1484–1500.
17. Yu B, Kumbier K (2020) Veridical data science. *Proceedings of the National Academy of Sciences* 117(8):3920–3929.
18. Agarwal A, Tan YS, Ronen O, Singh C, Yu B (2022) Hierarchical shrinkage: improving the accuracy and interpretability of tree-based methods. *arXiv preprint arXiv:2202.00858*.
19. Quinlan JR (1986) Induction of decision trees. *Machine learning* 1(1):81–106.
20. Lin J, Zhong C, Hu D, Rudin C, Seltzer M (2020) Generalized and scalable optimal sparse decision trees in *International Conference on Machine Learning*. (PMLR), pp. 6150–6160.
21. Hu X, Rudin C, Seltzer M (2019) Optimal sparse decision trees. *Advances in Neural Information Processing Systems (NeurIPS)*.
22. Bertsimas D, Dunn J (2017) Optimal classification trees. *Machine Learning* 106(7):1039–1082.
23. Pagallo G, Haussler D (1990) Boolean feature discovery in empirical learning. *Machine learning* 5(1):71–99.
24. Letham B, Rudin C, McCormick TH, Madigan D, , et al. (2015) Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics* 9(3):1350–1371.
25. Angelino E, Larus-Stone N, Alabi D, Seltzer M, Rudin C (2017) Learning certifiably optimal rule lists for categorical data. *arXiv preprint arXiv:1704.01701*.
26. Cohen WW, Singer Y (1999) A simple, fast, and effective rule learner. *AAAI/IAAI* 99(335-342):3.
27. Dembczyński K, Kotłowski W, Słowiński R (2008) Maximum likelihood rule ensembles in *Proceedings of the 25th international conference on Machine learning*. pp. 224–231.
28. Caruana R, et al. (2015) Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (ACM), pp. 1721–1730.
29. Breiman L, Friedman JH (1985) Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association* 80(391):580–598.
30. Friedman JH, Popescu BE, , et al. (2008) Predictive learning via rule ensembles. *The Annals of Applied Statistics* 2(3):916–954.
31. Friedman JH (1991) Multivariate adaptive regression splines. *The annals of statistics* pp. 1–67.
32. Freund Y, Schapire RE, , et al. (1996) Experiments with a new boosting algorithm in *icml*. (Citeseer), Vol. 96, pp. 148–156.
33. Chipman HA, George EI, McCulloch RE (2010) Bart: Bayesian additive regression trees. *The Annals of Applied Statistics* 4(1):266–298.
34. Luna JM, et al. (2019) Building more accurate decision trees with the additive tree. *Proceedings of the national academy of sciences* 116(40):19887–19893.
35. Lundberg SM, et al. (2019) Explainable ai for trees: From local explanations to global understanding. *arXiv preprint arXiv:1905.04610*.
36. Devlin S, Singh C, Murdoch WJ, Yu B (2019) Disentangled attribution curves for interpreting random forests and boosted trees. *arXiv preprint arXiv:1905.07631*.
37. Rudin C (2018) Please stop explaining black box models for high stakes decisions. *arXiv preprint arXiv:1811.10154*.
38. Wang T (2019) Gaining free or low-cost interpretability with interpretable partial substitute in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, eds. Chaudhuri K, Salakhutdinov R. (PMLR), Vol. 97, pp. 6505–6514.
39. Asuncion A, Newman D (2007) Uci machine learning repository.
40. Romano JD, et al. (2020) Pmlb v1. 0: an open source dataset collection for benchmarking machine learning methods. *arXiv preprint arXiv:2012.00058*.
41. Yeh IC, Lien Ch (2009) The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications* 36(2):2473–2480.
42. Osofsky JD (1997) The effects of exposure to violence on young children (1995). *Carnegie Corporation of New York Task Force on the Needs of Young Children; An earlier version of this article was presented as a position paper for the aforementioned corporation.*
43. Smith JW, Everhart JE, Dickson W, Knowler WC, Johannes RS (1988) Using the adap learning algorithm to forecast the onset of diabetes mellitus in *Proceedings of the annual symposium on computer application in medical care*. (American Medical Informatics Association), p. 261.
44. Pace RK, Barry R (1997) Sparse spatial autoregressions. *Statistics & Probability Letters* 33(3):291–297.
45. Nash WJ, Sellers TL, Talbot SR, Cawthorn AJ, Ford WB (1994) The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report* 48:p411.
46. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. *The Annals of statistics* 32(2):407–499.
47. Bertsimas D, Masiakos PT, Mylonas KS, Wiberg H (2019) Prediction of cervical spine injury in young pediatric patients: an optimal trees artificial intelligence approach. *Journal of Pediatric Surgery* 54(11):2353–2357.
48. Stiell IG, et al. (2001) The canadian ct head rule for patients with minor head injury. *The Lancet* 357(9266):1391–1396.
49. Kornblith AE, et al. (2022) Predictability and stability testing to assess clinical decision instrument performance for children after blunt torso trauma. *medRxiv*.
50. Holmes JF, et al. (2002) Identification of children with intra-abdominal injuries after blunt trauma. *Annals of emergency medicine* 39(5):500–509.
51. Kuppermann N, et al. (2009) Identification of children at very low risk of clinically-important brain injuries after head trauma: a prospective cohort study. *The Lancet* 374(9696):1160–1170.
52. Carreira-Perpinán MA, Tavallali P (2018) Alternating optimization of decision trees, with application to learning sparse oblique trees. *Advances in neural information processing systems* 31.
53. Leonard JC, et al. (2019) Cervical spine injury risk factors in children with blunt trauma. *Pediatrics* 144(1).
54. Raskutti G, J Wainwright M, Yu B (2012) Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research* 13(2).
55. Breiman L (1996) Bagging predictors. *Machine learning* 24:123–140.
56. Bühlmann P, Yu B (2002) Analyzing bagging. *The annals of Statistics* 30(4):927–961.
57. Mentch L, Zhou S (2020) Randomization as regularization: A degrees of freedom explanation for random forest success. *The Journal of Machine Learning Research* 21(1):6918–6953.
58. LeJeune D, Javadi H, Baraniuk R (2020) The implicit regularization of ordinary least squares ensembles in *International Conference on Artificial Intelligence and Statistics*. (PMLR), pp. 3525–3535.
59. Servén D, Brummitt C, Abedi H, hlink (2018) dswah/pygam: v0.8.0. *Zenodo*.
60. Schwarz G (1978) Estimating the dimension of a model. *The annals of statistics* pp. 461–464.
61. Lim C, Yu B (2016) Estimation stability with cross-validation (escv). *Journal of Computational and Graphical Statistics* 25(2):464–492.
62. Zuk O, Hechter E, Sunyaev SR, Lander ES (2012) The mystery of missing heritability: Genetic interactions create phantom heritability. *Proceedings of the National Academy of Sciences* 109(4):1193–1198.
63. Basu S, Kumbier K, Brown JB, Yu B (2018) iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences* p. 201711236.
64. Kumbier K, Basu S, Brown JB, Celniker S, Yu B (2018) Refining interaction search through signed iterative random forests. *arXiv preprint arXiv:1810.07287*.
65. Holmes JF, Lillis K, Monroe, David Borgialli D, Kerrey BT, , et al. (2013) Identifying children at very low risk of clinically important blunt abdominal injuries. *Annals of emergency medicine* 62(2):107–116.
66. Klusowski JM (2021) Universal consistency of decision trees in high dimensions. *arXiv preprint arXiv:2104.13881*.
67. Bennett P, Burch T, Miller M (1971) Diabetes mellitus in american (pima) indians. *The Lancet* 298(7716):125–128.
68. Meyer, Jr CD (1973) Generalized inversion of modified matrices. *Siam journal on applied mathematics* 24(3):315–323.

Supplement

The supplement contains further details of our investigation into FIGS. In Appendix S1, we discuss computational issues, and further possible extensions of the FIGS algorithm. In Appendix S2, we perform a number of synthetic simulations that examine how FIGS and Bagging-FIGS can adapt to a number of data generating processes in comparison to other methods. In Appendix S3, we provide empirical evidence of disentanglement by showing FIGS avoids repeated splits on a number of datasets. In Appendix S4, we provide an extended description of G-FIGS. Appendix S5 shows the CDIs learnt by G-FIGS for the IAI, TBI, and CSI datasets. Appendix S5 provides extended details on data cleaning for the the IAI, TBI, and CSI datasets, hyper-parameter selection for G-FIGS, as well as extended results for G-FIGS. Further, Appendix S5 includes a stability analysis of G-FIGS on the CSI dataset. Finally, Appendix S6 includes theoretical investigations into FIGS and tree-sum models.

S1. FIGS run-time analysis and extensions

FIGS run-time analysis. *The run time complexity for FIGS to grow a model with m splits in total is $O(dm^2n^2)$, where d the number of features, and n the number of samples.*

Proof. Each iteration of the outer loop adds exactly one split, so it suffices to bound the running time for each iteration, where it is clear that the cost is dominated by the operation `split` in Algorithm 1 line 9, which takes $O(n^2d)$, since there are at most nd possible splits, and it takes $O(n)$ time to compute the impurity decrease for each of these. Consider iteration s, in which we have a FIGS model f with s splits. Suppose f comprises k trees in total, with tree i having s_i splits, and so that $s = s_1 + \ldots + s_k$. The total number of potential splits is equal to $l + 1$, where l is the total number of leaves in the model. The number of leaves on each tree is $s_i + 1$, so the total number of leaves in f is

$$l = \sum_{i=1}^{k} (s_i + 1) = s + k.$$

Since each tree has at least one split, we have $k \leq s$, so that the number of potential splits is at most $2s + 1$ The total time complexity is therefore

$$\sum_{s=1}^{m} (2s + 1) \cdot O(n^2 d) = O(m^2 n^2 d).$$

□

FIGS extensions: Updating leaf values after tree structures are fixed. We can continue to update the leaf values of trees in the FIGS model after the stopping condition has been reached and the tree structures are fixed. To do this, we perform backfitting, i.e. we cycle through the trees in the model several times, and at each iteration, update the leaf values of a given tree to minimize the sum of squared residuals of the full model. This can be seen to be equivalent to block coordinate descent on a linear system, and so converges linearly, under regularity conditions, to the empirical risk minimizer among all functions that can be represented as sum of component functions which are each implementable by one of the tree structures.[¶¶] In our simulations, this postprocessing step usually does not seem to change the leaf values too much, probably because at the moment the leaves are created, their initial values are already chosen to minimize the mean-squared-error. Hence, we keep the step optional in our implementation of FIGS.

S2. FIGS Simulation Results.

We perform simulations to examine how well FIGS and Bagging-FIGS adapt to different data generating processes in comparison to four other methods: CART, random forests (RF), XGBoost (XGB), and generalized additive models (GAMs).

Generative models. Let $\mathbf{x}$ be a random variable with distribution π on $[0, 1]^d$. Here, we set $d = 50$, let $\epsilon \sim N(0, 0.01)$, with π uniform on $[0, 1]^{50}$ and investigate each of the following four regression functions:

(A) Linear model:
$f(\mathbf{x}) = \sum_{k=1}^{20} x_k + \epsilon$
(B) Single Boolean interaction model:
$f(\mathbf{x}) = \prod_{k=1}^{8} \mathbf{1}\{x_k > 0.1\} + \epsilon$
(C) Sum of polynomial interactions model:
$f(\mathbf{x}) = \sum_{k=1}^{5} x_{3k-2} x_{3k-1} x_{3k} + \epsilon$
(D) Local spiky sparse model (14):
$f(\mathbf{x}) = \sum_{k=1}^{5} \mathbf{1}\{x_{3k-2}, x_{3k-1}, x_{3k} > 0.5\} + \epsilon$

Performance metrics. For each choice of regression function, we vary the training set size in a geometrically-spaced grid between 100 and 2500. For each training set, we fit all five algorithms, and compute their noiseless test MSEs on a common test set of size 500. The entire experiment is repeated 10 times, and the results averaged across the replicates.

[¶¶] We say that a function is implementable by a tree structure if it is constant on each of the leaves of the tree.

Simulation results. FIGS and Bagging-FIGS predicts well across all four generative models, either performing best or very close to the best amongst all methods compared in moderate sample sizes (Fig S1). All other models suffer from weaknesses: PIRLS performs best for the linear generative model (A), but performs poorly whenever there are interactions present, i.e. for (B), (C) and (D). Tree ensembles (XGBoost and RF) perform well when there are interactions but fail to fit the linear generative model.

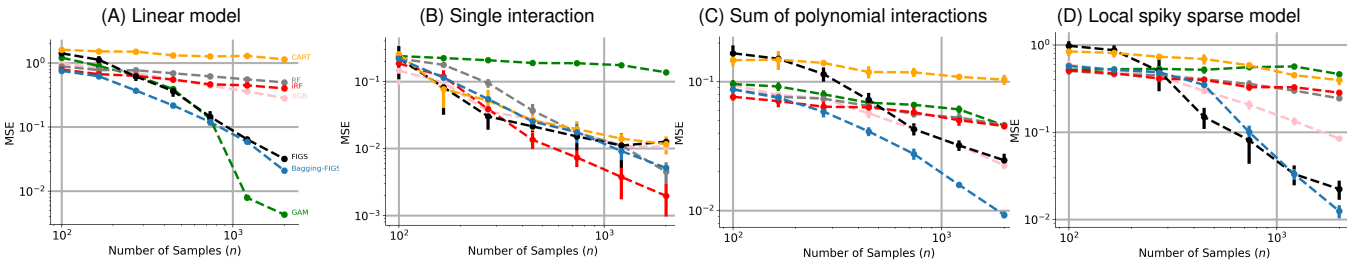


Fig. S1. FIGS is able to adapt to generative models, handling both additive structure and interactions gracefully. In both (A) and (B), the generative model for X is uniform with 50 features. Noise is Gaussian with mean zero and standard deviation 0.1 for training but no noise for testing. *(A)* Y is generated as linear model with sparsity 20 and coefficient 1. *(B)* Y is generated from a single Boolean interaction model of order 8. *(C)* Y is generated from a sum of 5 three-way polynomial interactions. *(D)* Y is generated from a sum of 5 three-way Boolean interactions. All results are averaged over 10 runs.

S3. Empirically learned number of trees and repeated splits for FIGS

Next, Fig S1 investigates whether FIGS avoids the issue of repeated splits. It shows the fraction of splits which are repeated within a learned model as a function of the total number of splits in the model. We define a split to be repeated if the model contains another split using the same feature and a threshold whose value is within 0.01 of the original split's threshold.*** FIGS consistently learns fewer repeated splits than CART, one signal that it is avoiding learning redundant subtrees by separately modeling additive components. Fig S1 shows he largest three datasets studied. Finally, Fig S2 shows the number of trees fitted by FIGS for each dataset.

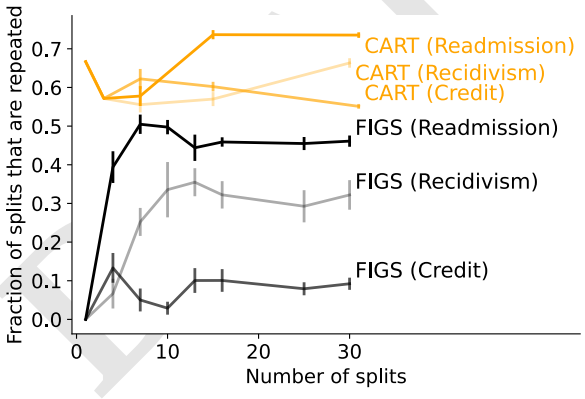


Fig. S1. FIGS learns less redundant models than CART. As a function of the number of splits in the learned model, we plot the fraction of splits repeated for the three largest datasets (all datasets studied show the same pattern). Error bars show standard error of the mean, computed over 6 random splits.

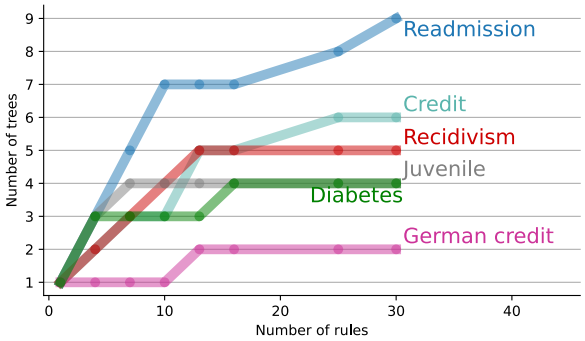


Fig. S2. Number of trees learned as a function of the total number of splits in FIGS for different classification datasets.

***This result is stable to reasonable variation in the choice of this threshold.

S4. G-FIGS Extended Description

Group probability-weighted FIGS (G-FIGS) aims to tackle two challenges: (1) sharing data across heterogenous groups in the input data and (2) ensuring the interpretability of the final output.

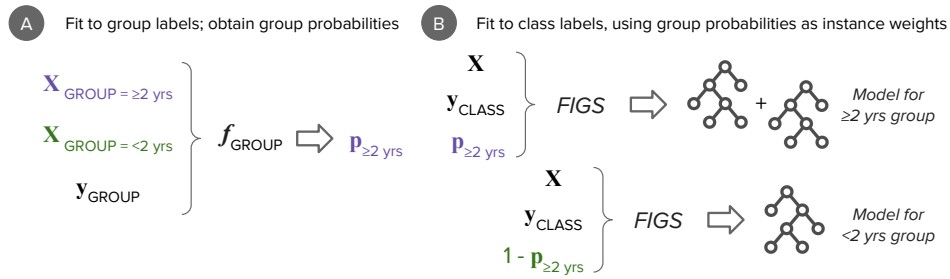


Fig. S1. Overview of G-FIGS. **(A)** First, the covariates of each instance in a dataset are used to estimate an instance-specific probability of membership in each of the pre-specified groups in the data (e.g., patients of age <2 years and ≥ 2 years). **(B)** Next, these membership probabilities are used as instance weights when fitting an interpretable model for each group.

Setup. We assume a supervised learning setting (classification or regression) with features X (e.g., *blood pressure*, *signs of vomiting*), and an outcome Y (e.g., *cervical spine injury*). We are also given a group label G, which is specified using the context of the problem and domain knowledge; for example, G may correspond to different sites at which data is collected, different demographic groups which are known to require different predictive models, or data before/after a key temporal event. G should be discrete, as G-FIGS will produce a separate model for each unique value of G, but may be a discretized continuous or count feature.

Fitting group membership probabilities. The first stage of G-FIGS fits a classifier to predict group membership probabilities $\mathbb{P}(G|X)$ (Fig S1A).[†††] Intuitively, these probabilities inform the degree to which a given instance is representative of a particular group; the larger the group membership probability, the more the instances should contribute to the model for that group. Any classifier can be used; we find that logistic regression and gradient-boosted decision trees perform best. The group membership probability classifier can be selected using cross-validation, either via group-label classification metrics or downstream performance of the weighted prediction model; we take the latter approach.

Fitting group probability-weighted FIGS. In the second stage (Fig S1B), for each group $G = g$, G-FIGS uses the estimated group membership probabilities, $\mathbb{P}(G = g|X)$, as instance weights in the loss function of a ML model for each group $\mathbb{P}(Y|X, G = g)$. Intuitively, this allows the outcome model for each group to use information from out-of-group instances when their covariates are sufficiently similar. While the choice of outcome model is flexible, we find that FIGS performs best when both interpretability and high predictive performance are required.[‡‡‡] By greedily fitting a sum of trees, FIGS effectively allocates a small budget of splits to different types of structure in data.

S5. CDI Results

A. Fitted clinical-decision instruments. The fitted clinical-decision instruments (CDIs) for the IAI, TBI, and CSI datasets from PECARN are shown in Fig S1, Fig S2, and Fig S3 respectively.

[†††] In estimating $\mathbb{P}(G = g|X)$, we exclude features that trivially identify G (e.g., we exclude age when values of G are age ranges).

[‡‡‡] When interpretability is not critical, the same weighting procedure could also be applied to black-box models, such as Random Forest (1).

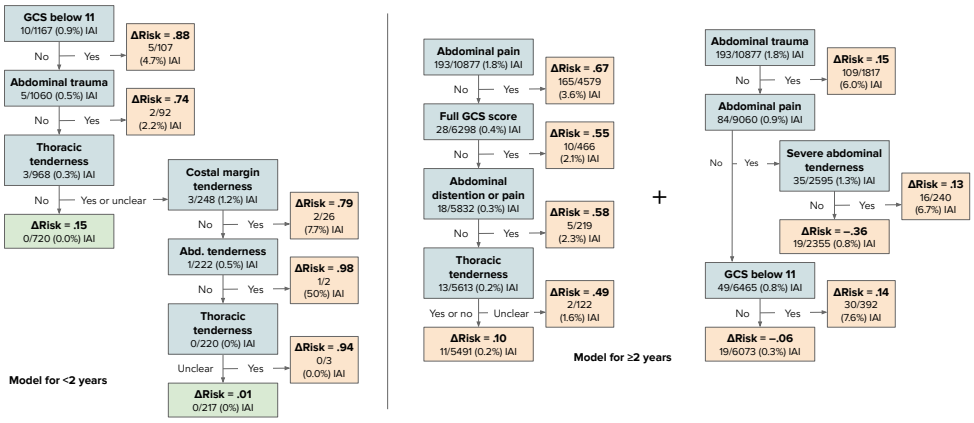


Fig. S1. G-FIGS model fitted to the IAI dataset. Note that the younger group only uses *tenderness*, which can evaluated without verbal input from the patient, whereas the older group uses *pain*, which requires a verbal response. Achieves 95.1% sensitivity and 50.8% specificity (training).

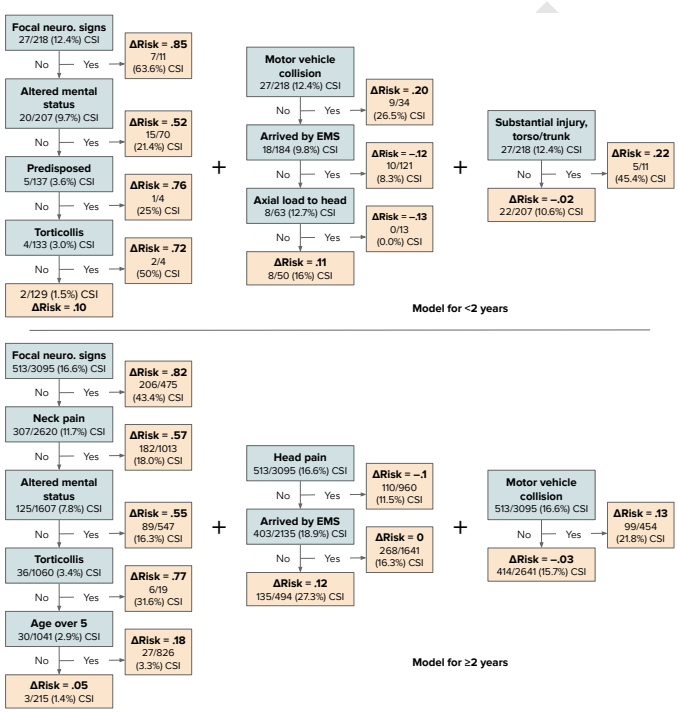


Fig. S3. G-FIGS model fitted to the CSI dataset. Achieves 97.0% sensitivity and 33.9% specificity (training). The left tree for <2 years gives large Δ Risk to active features, and on its own provides sensitivity of 99%. Counterintuitively, the middle tree assigns Δ Risk < 0 for patients arriving by ambulance (*EMS*) or with head injuries that affect the spine (*axial load*). However, adding this second tree results in boosted specificity (increase of 8.7%) with a tiny reduction in sensitivity (decrease of 0.4%), indicating that G-FIGS adaptively tunes the sensitivity-specificity tradeoff.

B. Interpreting the group-membership model. Recall that fitting G-FIGS requires two steps (see Appendix S4): (1) fitting a group-membership model to obtain sample weights and then (2) using these sample weights to fit a FIGS model. In this section, we interpret the fitted group-membership models. In the clinical context, we begin by fitting several logistic regression and gradient-boosted decision tree group membership models to each of the training datasets (Table 2) to predict whether a patient is in the <2 years or ≥ 2 years age group. For the instance-weighted methods, we treat the choice of group membership model as a hyperparameter, and select the best model according to the downstream performance of the final decision instrument on the validation set.

Table S1 shows the coefficients of the most important features for each logistic regression group membership model when predicting whether a patient is in the ≥ 2 years age group. The coefficients reflect existing medical expertise. For example, the presence of verbal response features (e.g., *Amnesia*, *Headache*) increases the probability of being in the ≥ 2 years group, as does the presence of activities not typical for the <2 years age group (e.g. *Bike injury*).

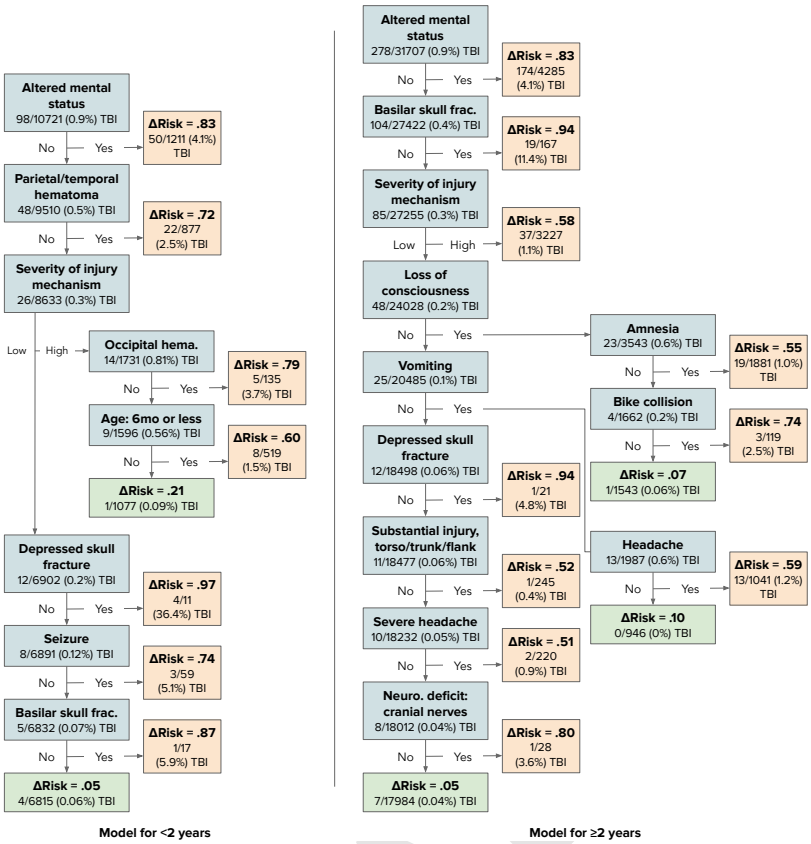


Fig. S2. G-FIGS model fitted to the TBI dataset. Interestingly, in this case G-FIGS learns only a single tree for each group. Note that the model for the older group utilizes the *Headache* and *Severe Headache* features, which require a verbal response. Achieves 97.1% sensitivity and 58.9% specificity (training).

C. Extended clinical-decision results. In Table S2 and Table S3, we report the results of G-FIGS and other baselines for all data-sets for the < 2 and > 2 years age group separately. As seen in the results, we see that G-FIGS improve upon the baseline methods and FIGS particularly at high levels of sensitivities. Additionally, in Table S4, we include the results from above with their standard errors, as well as additional metrics (AUC and F1 score) for each dataset.

Traumatic brain injury		Cervical spine injury		Intra-abdominal injury	
Variable	Coefficient	Variable	Coefficient	Variable	Coefficient
No fontanelle bulging	3.62	Neck tenderness	2.44	Bike injury	2.01
Amnesia	2.07	Neck pain	2.18	Abdomen pain	1.66
Pedestrian struck by vehicle	1.44	Motor vehicle injury: other	1.54	Thoracic tenderness	1.43
Headache	1.39	Hit by car	1.47	Hypotension	1.23
Bike injury	1.26	Substantial injury: extremity	1.35	No abdomen pain	0.98

Table S1. Logistic regression coefficients for features that contribute to high $\mathbb{P}$(age ≥ 2 years $\mid X$) reflect known medical knowledge. For example, features with large coefficients require verbal responses (e.g., *Amnesia*, *Headache*, *Pain*), relate to activities not typical for the <2 years age group (*Bike injury*), or are specific to older children, e.g., older children should have *No fontanelle bulging*, as cranial soft spots typically close by 2 to 3 months after birth.

	Cervical spine injury				Traumatic Brain Injury				Intra-abdominal injury			
Sensitivity level:	92%	94%	96%	98%	92%	94%	96%	98%	92%	94%	96%	98%
TAO	45.8	45.8	45.8	45.8	7.7	7.7	0.0	0.0	33.3	33.3	33.3	6.7
TAO-SEP	0.0	0.0	0.0	0.0	20.6	14.3	8.3	8.3	0.0	0.0	0.0	0.0
CART	45.8	45.8	45.8	45.8	19.0	19.0	7.1	1.2	29.6	29.6	29.6	29.6
CART-SEP	0.0	0.0	0.0	0.0	20.6	14.3	8.3	8.3	0.0	0.0	0.0	0.0
G-CART	36.4	36.4	36.4	36.4	16.1	15.6	8.7	**8.7**	4.4	4.4	4.4	4.4
FIGS	56.1	56.1	56.1	56.1	**36.3**	**30.3**	5.9	0.1	**39.4**	**39.4**	**39.4**	**39.4**
FIGS-SEP	6.9	6.9	6.9	6.9	31.1	25.1	13.0	6.7	9.0	9.0	9.0	9.0
G-FIGS	**65.9**	**65.9**	**65.9**	**65.9**	17.2	17.2	**13.7**	7.5	24.3	24.3	24.3	24.3

Table S2. < 2 years age group test set prediction results averaged over 10 random data splits. Values in columns labeled with a sensitivity percentage (e.g. 92%) are best specificity achieved at the given level of sensitivity or greater.

	Cervical spine injury				Traumatic Brain Injury				Intra-abdominal injury			
Sensitivity level:	92%	94%	96%	98%	92%	94%	96%	98%	92%	94%	96%	98%
TAO	39.5	20.1	0.2	0.2	12.2	6.1	6.1	0.3	0.3	0.3	0.0	0.0
TAO-SEP	33.6	17.9	3.5	1.4	19.8	13.4	7.4	0.5	10.2	5.5	4.2	1.3
CART	22.0	15.4	14.8	2.1	19.0	19.0	7.1	1.2	7.6	2.8	1.6	1.3
CART-SEP	33.0	17.3	2.8	1.4	19.7	13.4	7.3	0.8	9.0	4.2	4.2	1.3
G-CART	37.3	16.2	1.9	1.7	19.6	7.2	7.2	0.8	14.7	10.5	3.9	0.7
FIGS	37.8	33.6	**25.5**	**14.1**	24.5	18.1	18.1	0.5	24.9	14.6	1.4	0.0
FIGS-SEP	39.5	**33.8**	22.0	11.6	25.3	20.3	18.7	**6.3**	**28.0**	19.0	9.2	0.5
G-FIGS	**40.7**	33.5	23.8	13.5	**44.1**	**31.5**	**19.5**	5.3	27.9	**22.1**	**13.8**	**2.4**

Table S3. > 2 years age group test set prediction results averaged over 10 random data splits. Values in columns labeled with a sensitivity percentage (e.g. 92%) are best specificity achieved at the given level of sensitivity or greater.

	Traumatic brain injury						Cervical spine injury		
	92%	94%	96%	98%	ROC AUC	F1	92%	94%	96%
TAO	6.2 (5.9)	6.2 (5.9)	0.4 (0.4)	0.4 (0.4)	.294 (.05)	5.2 (.00)	41.5 (0.9)	21.2 (6.6)	0.2 (0.2)
TAO-SEP	26.7 (6.4)	13.9 (5.4)	10.4 (5.5)	2.4 (1.5)	.748 (.02)	**5.8** (.00)	32.5 (4.9)	7.0 (1.6)	5.4 (0.7)
CART	20.9 (8.8)	14.8 (7.6)	7.8 (5.8)	2.1 (0.6)	.702 (.06)	5.7 (.00)	38.6 (3.6)	13.7 (5.7)	1.5 (0.6)
CART-SEP	26.6 (6.4)	13.8 (5.4)	10.3 (5.5)	2.4 (1.5)	.753 (.02)	5.6 (.00)	32.1 (5.1)	7.8 (1.5)	5.4 (0.7)
G-CART	15.5 (5.5)	13.5 (5.7)	6.4 (2.2)	3.0 (1.5)	**.758** (.01)	5.5 (.00)	38.5 (3.4)	15.2 (4.8)	4.9 (1.0)
FIGS	23.8 (9.0)	18.2 (8.5)	12.1 (7.3)	0.4 (0.3)	.380 (.07)	4.8 (.00)	39.1 (3.0)	33.8 (2.4)	24.2 (3.2)
FIGS-SEP	39.9 (7.9)	19.7 (6.8)	**17.5** (7.0)	2.6 (1.6)	.619 (.05)	5.1 (.00)	38.7 (1.6)	33.1 (2.0)	20.1 (2.6)
G-FIGS	**42.0** (6.6)	**23.0** (7.8)	14.7 (6.5)	**6.4** (2.8)	.696 (.04)	4.7 (.00)	**42.2** (1.3)	**36.2** (2.3)	**28.4** (3.8)

	Cervical spine injury (cont.)			Intra-abdominal injury					
	98%	ROC AUC	F1	92%	94%	96%	98%	ROC AUC	F1
TAO	0.2 (0.2)	.422 (.04)	44.5 (.01)	0.2 (0.2)	0.2 (0.2)	0.0 (0.0)	0.0 (0.0)	.372 (.04)	**13.9** (.01)
TAO-SEP	2.5 (1.0)	.702 (.01)	44.4 (.01)	12.1 (1.7)	8.5 (2.0)	2.0 (1.3)	0.0 (0.0)	.675 (.01)	12.9 (.00)
CART	1.1 (0.4)	.617 (.06)	**45.8** (.01)	11.8 (5.0)	2.7 (1.0)	1.6 (0.5)	1.4 (0.5)	.688 (.06)	13.4 (.00)
CART-SEP	2.5 (1.0)	.707 (.00)	44.2 (.01)	11.0 (1.6)	9.3 (1.8)	2.8 (1.4)	0.0 (0.0)	.688 (.01)	13.0 (.01)
G-CART	3.9 (1.1)	**.751** (.01)	45.2 (.01)	11.7 (1.3)	10.1 (1.6)	3.8 (1.3)	0.7 (0.4)	**.732** (.02)	12.5 (.01)
FIGS	**16.7** (3.9)	.664 (.03)	43.0 (.01)	**32.1** (5.5)	13.7 (6.0)	1.4 (0.8)	0.0 (0.0)	.541 (.04)	9.4 (.01)
FIGS-SEP	3.9 (2.2)	.643 (.02)	41.4 (.01)	18.8 (4.4)	9.2 (2.2)	2.6 (1.7)	0.9 (0.8)	.653 (.02)	8.0 (.00)
G-FIGS	15.7 (3.9)	.700 (.01)	42.6 (.01)	29.7 (6.9)	**18.8** (6.6)	**11.7** (5.1)	**3.0** (1.3)	.671 (.03)	9.1 (.01)

Table S4. Test set prediction results averaged over 10 random data splits, with corresponding standard error in parentheses. Values in columns labeled with a sensitivity percentage (e.g. 92%) are best specificity achieved at the given level of sensitivity or greater. G-FIGS provides the best performance overall in the high-sensitivity regime. G-CART attains the best ROC curves, while TAO is strongest in terms of F1 score.

D. Clinical data-preprocessing details.

Traumatic brain injury (TBI) To screen patients, we follow the inclusion and exclusion criteria from previous work (51), which excludes patients with Glasgow Coma Scale (GCS) scores under 14 or no signs or symptoms of head trauma, among other disqualifying factors. No

patients were dropped due to missing values: the majority of patients have about 1% of features missing, and are at maximum still under 20%. We utilize the same set of features as a previous study (51).

Our strategy for imputing missing values differed between features according to clinical guidance. For features that are unlikely to be left unrecorded if present, such as paralysis, missing values were assumed to be negative. For other features that could be unnoticed by clinicians or guardians, such as loss of consciousness, missing values are assumed to be positive. For features that did not fit into either of these groups or were numeric, missing values are imputed with the median.

Cervical spine injury (CSI) (53) engineered a set of 22 expert features from 609 raw features; we utilize this set but add back features that provide information on the following:

- Patient position after injury
- Clinical intervention received by patients prior to arrival (immobilization, intubation)
- Pain and tenderness of the head, face, torso/trunk, and extremities
- Age and gender
- Whether the patient arrived by emergency medical service (EMS)

We follow the same imputation strategy described in the TBI paragraph above. Features that are assumed to be negative if missing include focal neurological findings, motor vehicle collision, and torticollis, while the only feature assumed to be positive if missing is loss of consciousness.

Intra-abdominal injury (IAI) We follow the data preprocessing steps described in (65) and (49). In particular, all features of which at least 5% of values are missing are removed, and variables that exhibit insufficient inter-rater agreement (lower bound of 95% CI under 0.4) are removed. The remaining missing values are imputed with the median. In addition to the 18 original variables, we engineered three additional features:

- *Full GCS score*: True when GCS is equal to the maximum score of 15
- *Abd. Distention or abd. pain*: Either abdominal distention or abdominal pain
- *Abd. trauma or seatbelt sign*: Either abdominal trauma or seatbelt sign

Data for predicting group membership probabilities The data preprocessing steps for the group membership models in the first step of G-FIGS are identical to that above, except that missing values are not imputed at all for categorical features, such that "missing", or NaN, is allowed as one of the feature labels in the data. We find that this results in more accurate group membership probabilities, since for some features, such as those requiring a verbal response, missing values are predictive of age group.

Unprocessed data is available at https://pecarn.org/datasets/ and clean data is available on github at https://github.com/csinva/imodels-data (easily accessibly through the imodels package (7)).

Traumatic brain injury		
Feature Name	% Missing	% Nonzero
Altered Mental Status	0.74	12.95
Altered Mental Status: Agitated	87.05	1.79
Altered Mental Status: Other	87.05	1.82
Altered Mental Status: Repetitive	87.05	1.04
Altered Mental Status: Sleepy	87.05	6.67
Altered Mental Status: Slow to respond	87.05	3.22
Acting normally per parents	7.09	85.38
Age (months)	0.00	N/A
Verbal amnesia	38.41	10.45
Trauma above clavicles	0.30	64.38
Trauma above clavicles: Face	35.92	29.99
Trauma above clavicles: Scalp-frontal	35.92	20.48
Trauma above clavicles: Neck	35.92	1.38
Trauma above clavicles: Scalp-occipital	35.92	9.62
Trauma above clavicles: Scalp-parietal	35.92	7.79
Trauma above clavicles: Scalp-temporal	35.92	3.39
Drugs suspected	4.19	0.87
Fontanelle bulging	0.37	0.06
Sex	0.01	N/A
Headache severity	2.38	N/A
Headache start time	3.09	N/A
Headache	32.76	29.94
Hematoma	0.69	39.42
Hematoma location	0.47	N/A
Hematoma size	1.67	N/A
Severity of injury mechanism	0.74	N/A
Injury mechanism	0.67	N/A
Intubated	0.73	0.01
Loss of consciousness	4.05	10.37
Length of loss of consciousness	5.39	N/A
Neurological deficit	0.85	1.3
Neurological deficit: Cranial	98.70	0.18
Neurological deficit: Motor	98.70	0.28
Neurological deficit: Other	98.70	0.71
Neurological deficit: Reflex	98.70	0.03
Neurological deficit: Sensory	98.70	0.26
Other substantial injury	0.43	10.07
Other substantial injury: Abdomen	89.93	1.25
Other substantial injury: Cervical spine	89.93	1.37
Other substantial injury: Cut	89.93	0.12
Other substantial injury: Extremity	89.93	5.49
Other substantial injury: Flank	89.93	1.56
Other substantial injury: Other	89.93	1.65
Other substantial injury: Pelvis	89.93	0.44
Paralyzed	0.75	0.01
Basilar skull fracture	0.99	0.68
Basilar skull fracture: Hemotympanum	99.32	0.35
Basilar skull fracture: CSF otorrhea	99.32	0.04
Basilar skull fracture: Periorbital	99.32	0.19
Basilar skull fracture: Retroauricular	99.32	0.08
Basilar skull fracture: CSF rhinorrhea	99.32	0.03
Skull fracture: Palpable	0.24	0.38
Skull fracture: Palpable and depressed	99.69	0.18
Sedated	0.76	0.08
Seizure	1.70	1.17
Length of seizure	0.18	N/A
Time of seizure	0.12	N/A
Vomiting	0.71	13.1
Time of last vomit	89.04	N/A
Number of times vomited	0.60	N/A
Vomit start time	0.87	N/A
Intra-abdominal injury		
Abdominal distention	4.38	2.3
Abdominal distention or pain	0.00	4.93
Degree of abdominal tenderness	70.13	N/A
Abdominal trauma	0.56	15.48
Abdominal trauma or seat belt sign	0.00	16.3
Abdomen pain	15.38	30.06
Age (years)	0.00	N/A
Costal margin tenderness	0.00	11.33
Decreased breath sound	1.93	2.13
Distracting pain	7.38	23.29
Glasgow Coma Scale (GCS) score	0.00	N/A
Full GCS score	0.00	86.21
Hypotension	0.00	1.44
Left costal margin tenderness	0.00	N/A
Method of injury	3.95	N/A
Right costal margin tenderness	0.00	N/A
Seat belt sign	3.30	4.93
Sex	0.00	N/A
Thoracic tenderness	9.99	15.96
Thoracic trauma	0.63	16.95
Vomiting	3.92	9.57
Cervical spine injury		
Age (years)	0.00	N/A
Altered mental status	2.05	24.72
Axial load to head	0.00	24.0
Clotheslining	3.38	0.94
Focal neurological findings	9.84	14.67
Method of injury: Diving	0.03	1.3
Method of injury: Fall	2.44	3.83
Method of injury: Hanging	0.03	0.15
Method of injury: Hit by car	0.03	15.09
Method of injury: Auto collision	7.73	14.73
Method of injury: Other auto	0.03	3.11
Arrived by EMS	0.00	77.24
Loss of consciousness	8.03	42.68
Neck pain	5.25	38.42
Posterior midline neck tenderness	2.57	29.88
Patient position on arrival	61.52	N/A
Predisposed	0.00	0.66
Pain: Extremity	18.35	25.87
Pain: Face	18.35	7.58
Pain: Head	18.35	29.04
Pain: Torso/trunk	18.35	28.95
Tenderness: Extremity	20.37	15.15
Tenderness: Face	20.37	3.83
Tenderness: Head	20.37	7.79
Tenderness: Torso/trunk	20.37	25.87
Substantial injury: Extremity	1.03	10.87
Substantial injury: Face	1.06	5.67
Substantial injury: Head	1.00	15.88
Substantial injury: Torso/trunk	1.03	7.3
Neck tenderness	2.48	39.3
Torticollis	7.03	5.77
Ambulatory	5.77	21.46
Axial load to top of head	0.00	2.35
Sex	0.00	N/A

Table S5. Final features used for fitting the *outcome* models. Features include information about patient history (i.e. *mechanism of injury*), physical examination (i.e. *Abdominal trauma*), and mental condition (i.e. *Altered mental status*). Percentage of nonzero values is marked *N/A* for non-binary features.

E. Clinical-data hyperparameter selection.

Data splitting We use 10 random training/validation/test splits for each dataset, performing hyperparameter selection separately on each. There are two reasons we choose not to use a fixed test set. First, the small number of positive instances in our datasets makes our primary metrics (specificity at high sensitivity levels) noisy, so averaging across multiple splits makes the results more stable. Second, the works that introduced the TBI, IAI, and CSI datasets did not publish their test sets, as it is not as common to do so in the medical field as it is in machine learning, making the choice of test set unclear. For TBI and CSI, we simply use the random seeds 0 through 10. For IAI, some filtering of seeds is required due to the low number of positive examples; we reject seeds that do not allocate positive examples evenly enough between each split (a ratio of negative to positive outcomes over 200 in any split).

Class weights Due to the importance of achieving high sensitivity, we upweight positive instances in the loss by the inverse proportion of positive instances in the dataset. This results in class weights of about 7:1 for CSI, 112:1 for TBI, and 60:1 for IAI. These weights are fixed for all methods.

Hyperparameter settings Due to the relatively small number of positive examples in all datasets, we keep the hyperparameter search space small to avoid overfitting. We vary the maximum number of tree splits from 8 to 16 for all methods and the maximum number of update iterations from 1 to 5 for TAO. The options of group membership model are logistic regression with L2 regularization and gradient-boosted trees (2). For both models, we simply include two hyperparameter settings: a less-regularized version and a more-regularized version, by varying the inverse regularization strength (C) for logistic regression and the number of trees (N) for gradient-boosted trees. We initially experimented with random forests and CART, but found them to lead to poor downstream performance. Random forests tended to separate the groups too well in terms of estimated probabilities, leading to little information sharing between groups, while CART did not provide unique enough membership probabilities, since CART probability estimates are simply within-node class proportions.

Validation metrics We use the highest specificity achieved when sensitivity is at or above 94% as the metric for validation. If this metric is tied between different hyperparameter settings of the same model, specificity at 90% sensitivity is used as the tiebreaker. For the IAI dataset, only specificity at 90% sensitivity is used, since the relatively small number of positive examples makes high sensitivity metrics noisier than usual. If there is still a tie at 90% sensitivity, the smaller model in terms of number of tree splits is chosen.

Validation of group membership model Hyperparameter selection for G-FIGS and G-CART is done in two stages due to the need to select the best group membership model. First, the best-performing maximum of tree splits is selected for each combination of method and membership model. This is done separately for each data group. Next, the best membership model is selected using the overall performance of the best models across both data groups. The two-stage validation process ensures that the <2 years and ≥ 2 years age groups use the same group membership probabilities, which we have found performs better than allowing different sub-models of G-FIGS to use different membership models.

	<2 years group			≥2 years group		
Maximum tree splits:	8	12	16	8	12	16
TAO (1 iter)	**15.1** (6.7)	15.1 (6.7)	14.4 (6.1)	**14.1** (7.8)	14.1 (7.8)	8.9 (5.9)
TAO (5 iter)	**14.4** (6.1)	0.0 (0.0)	0.0 (0.0)	**8.9** (5.9)	3.1 (0.9)	1.5 (0.7)
CART-SEP	**15.1** (6.7)	14.4 (6.1)	0.0 (0.0)	**14.0** (7.8)	8.9 (5.9)	3.1 (0.9)
FIGS-SEP	**13.7** (5.9)	0.0 (0.0)	0.0 (0.0)	**23.1** (8.8)	13.0 (7.4)	7.8 (5.6)
G-CART w/ LR ($C = 2.8$)	**7.9** (6.7)	3.1 (2.1)	3.5 (1.7)	19.0 (8.8)	**21.8** (8.4)	2.1 (0.6)
G-CART w/ LR ($C = 0.1$)	**20.4** (8.6)	8.3 (6.6)	10.1 (6.7)	12.7 (7.6)	**14.9** (7.1)	3.6 (0.9)
G-CART w/ GB ($N = 100$)	**19.8** (8.3)	7.2 (6.3)	7.6 (6.1)	13.3 (8.0)	**21.4** (8.5)	9.0 (5.6)
G-CART w/ GB ($N = 50$)	**26.8** (9.7)	8.1 (6.3)	8.4 (6.1)	13.3 (8.0)	**21.4** (8.5)	9.7 (5.6)
G-FIGS w/ LR ($C = 2.8$)	**14.9** (8.5)	7.5 (5.4)	8.1 (6.9)	41.0 (8.7)	**48.1** (8.2)	35.6 (8.9)
G-FIGS w/ LR ($C = 0.1$)	**31.0** (9.4)	23.1 (9.1)	25.9 (9.7)	46.9 (8.4)	**48.2** (8.4)	33.7 (8.9)
G-FIGS w/ GB ($N = 100$)	**24.5** (8.6)	24.0 (9.3)	21.2 (8.7)	**47.5** (8.5)	47.5 (8.2)	27.9 (8.6)
G-FIGS w/ GB ($N = 50$)	**32.1** (9.6)	18.3 (8.2)	12.7 (6.9)	47.5 (8.5)	**53.2** (7.3)	28.4 (8.3)

(a)

Group membership model:	LR ($C = 2.8$)	LR ($C = 0.1$)	GB ($N = 100$)	GB ($N = 50$)
G-CART (<2 years, ≥2 years models combined)	**27.8** (6.0)	21.5 (5.9)	19.0 (5.7)	27.1 (6.5)
G-FIGS (<2 years, ≥2 years models combined)	51.3 (5.8)	54.5 (6.2)	**57.4** (5.6)	44.6 (7.4)

(b)

Table S6. Hyperparameter selection table for the TBI dataset; the metric shown is specificity at 94% sensitivity on the validation set, with corresponding standard error in parentheses. First, the best-performing maximum of tree splits is selected for each method or combination of method and membership model (a). This is done separately for each data group. Next, the best membership model is selected for G-CART and G-FIGS using the overall performance of the best models from (a) across both data groups (b). The two-stage validation process ensures that the <2 years and ≥ 2 years age groups use the same group membership probabilities, which we have found leads to better performance than allowing them to use different membership models. Metrics shown are averages across the 10 validation sets, but hyperparameter selection was done independently for each of the 10 data splits.

F. PCS analysis of CSI G-FIGS model. In this section, we perform a stability analysis of the G-FIGS fitted on the CSI dataset. As discussed earlier, stability is a crucial pre-requisite for using ML models to interpret real-world scientific problems. To measure the stability of G-FIGS, we artificially introduce noise by randomly swapping a percentage p of labels $\mathbf{y}$. We vary p between $\{1\%, 2.5\%, 5\%\}$. For each value of p, we measure stability by comparing the similarity of the feature sets selected in the model trained on the perturbed data to the model displayed in Fig 3. The similarity of feature sets is measured via the Jaccard distance. That is, let $\hat{f}$ denote the FIGS model fitted on the unperturbed data. Further, define $\hat{S}$ as the features split on in $\hat{f}$. Similarly, let $\hat{f}^p$ denote the FIGS fitted on the perturbed data. Note that $\hat{f}^p$ does not necessarily need to consist of the same number of trees as $\hat{f}$. Moreover, we define $\hat{S}^p$ as the features split on in $\hat{f}^p$. Then, we define the stability score of $\hat{f}$ as follows

$$\text{Sta}(\hat{f}) = \frac{\hat{S} \cap \hat{S}^p}{\hat{S} \cup \hat{S}^p} \quad [1]$$

For G-FIGS we compute the stability score of the model fitted on each age group. We measure the stability score of G-FIGS over 5 repetitions, and display the results in Table S7.

percentage of labels swapped p:	1%	2.5%	5%
Sta(G-FIGS (<2 years))	1.0	0.98	0.93
Sta(G-FIGS (>2 years))	0.86	0.64	0.64

Table S7. Stability analysis results for G-FIGS fitted on the CSI dataset when a percentage p of labels are flipped. We vary p betweem $\{1\%, 2.5\%, 5\%\}$. The stability of G-FIGS is measured via Eq. (1)**, and we average our results over over 5 repetitions. The results indicate G-FIGS is stable to perturbations, in particular for the <2 years age group.**

S6. Theoretical Results

In this section, we discuss our theoretical results relating to FIGS and tree-sum models.

A. CART as local orthogonal greedy procedure. In this section, we build on recent work which shows that CART can be thought of as a "local orthogonal greedy procedure" (66). To see this, consider a tree model $\hat{f}$, and a leaf node $\mathtt{t}$ in the tree. Given a potential split s of $\mathtt{t}$ into

children t_L and t_R, we may associate the normalized decision stump

$$\psi_{\mathrm{t},s}(\mathbf{x}) = \frac{N(\mathrm{t}_R)\mathbf{1}\{\mathbf{x} \in \mathrm{t}_L\} - N(\mathrm{t}_L)\{\mathbf{x} \in \mathrm{t}_R\}}{\sqrt{N(\mathrm{t})N(\mathrm{t}_L)N(\mathrm{t}_R)}}, \quad [2]$$

where $N(-)$ is used to denote the number of samples in a given node. We use $\mathbf{\Psi}_{\mathrm{t},s}$ to denote the vector in $\mathbb{R}^n$ comprising its values on the training set, noticing that it has unit norm. If t is an interior node, then there is already a designated split $s(\mathrm{t})$, and we drop the second part of the subscript. It is easy to see that the collection $\{\mathbf{\Psi}_{\mathrm{t}}\}_{\mathrm{t}\in\hat{f}}$ is orthogonal to each other, and also to all decision stumps associated to potential splits. This gives the second equality in the following chain

$$\hat{\Delta}(s,\mathrm{t}) = (\mathbf{y}^T\mathbf{\Psi}_{\mathrm{t},s})^2 = (\mathbf{r}^T\mathbf{\Psi}_{\mathrm{t},s})^2, \quad [3]$$

with the first being a straightforward calculation. As such, the CART splitting condition is equivalent to selecting a feature vector from an admissible set that best reduces the residual variance. The CART update rule adds this feature to the model, and sets its coefficient to minimize the 1-dimensional least squares equation. Given orthogonality of all the features, this also updates the linear model to the best fit linear model on the new feature set.

Concatenating the decision stumps together yields a feature map $\Psi\colon \mathbb{R}^d \to \mathbb{R}^p$, and we let $\mathbf{\Psi}$ denote the n by m transformed data matrix. Let $\hat{\boldsymbol{\beta}}$ denote the solution to the least squares problem

$$\min_{\boldsymbol{\beta}} \|\mathbf{\Psi}\boldsymbol{\beta} - \mathbf{y}\|_2^2. \quad [4]$$

We have just argued that we have functional equality

$$\hat{f}(\mathbf{x}) = \hat{\boldsymbol{\beta}}^T\Psi(\mathbf{x}). \quad [5]$$

These calculations can be found in more detail in Lemma 3.2 in (66).

B. Modifications for FIGS. With a collection of trees $\mathcal{T}_1, \ldots, \mathcal{T}_K$, we may still associate a normalized decision stump Eq. (2) to every node and every potential split. The impurity decrease used to determine splits can still be written as a squared correlation with a potential split vector,

$$\hat{\Delta}(s,\mathrm{t},r) = (\mathbf{r}^{(-k)T}\mathbf{\Psi}_{\mathrm{t},s})^2.$$

and so the FIGS splitting rule can also be thought of as selecting a feature vector from an admissible set that best reduces the residual variance, while its update rule adds this feature to the model, and sets its coefficient to minimize the 1-dimensional least squares equation. The difference to CART lies in the fact that the admissible set is now larger, comprising potential splits from multiple trees, and furthermore, the node vectors from different trees are no longer orthogonal to each other, and so the update rule, while solving the 1-dimensional problem, no longer minimizes the full least squares loss. Nonetheless, as discussed in the main paper, a simple backfitting procedure after the tree structures are fixed is equivalent to block coordinate descent on this linear system, and converges to the minimizer, and furthermore, we observed in our examples that the resulting coefficients do not change too much from their initial values, meaning that the FIGS solution is already close to a best fit linear model.

C. FIGS disentangles the additive components of additive generative models.

Generative model. Let $\mathbf{x}$ be a random variable with distribution π on $[0,1]^d$. Suppose that we have disjoint blocks of features $I_1, \ldots, I_K$, of sizes $d_1, \ldots, d_K$, with $d = \sum_{k=1}^K d_k$, and suppose the blocks of features are mutually independent, i.e. $\mathbf{x}_{I_j} \perp\!\!\!\perp \mathbf{x}_{I_k}$ for $j \neq k$, where for any index set I, $\mathbf{x}_I$ denotes the subvector of $\mathbf{x}$ comprising coordinates in I. Let $y = f(\mathbf{x}) + \epsilon$ where $\mathbb{E}\{\epsilon \mid \mathbf{x}\} = 0$ and

$$f(\mathbf{x}) = \sum_{k=1}^K f_k(\mathbf{x}_{I_k}). \quad [6]$$

Suppose further that each component function has mean zero with respect to π.

For a given tree structure $\mathcal{T}$, let $J(\mathcal{T})$ denote the set of features that it splits upon. We say that a tree-sum model with tree structures $\{\mathcal{T}_1, \ldots, \mathcal{T}_M\}$ completely (respectively partially) *disentangles* the additive components of an additive generative model Eq. (6) if $M = K$ and, after re-indexing if necessary, $J(\mathcal{T}_k) = I_k$ (respectively $J(\mathcal{T}_k) \subset I_k$) for $k = 1, \ldots, K$.

In this section, we argue that FIGS is able to achieve disentanglement and learn additive components of additive generative models. To the best of our knowledge, this property is unique to FIGS and is not shared by any other tree ensemble algorithm. As argued in the introduction, disentanglement helps to avoid duplicate subtrees, leading to a more parsimonious model with better generalization performance (see Theorem 2 for a precise statement) . Note that even when the generative model is not additive, FIGS helps to reduce the number of possibly redundant often observed in CART models (see Appendix S2.)

Theorem 1 (Partial disentanglement in large sample limit). *Consider the generative model* Eq. (6)*, and recall the notation from Sec 2. Suppose we run Algorithm 1 with the following oracle modifications when splitting a node* t*.*

1. Splits are selected using the population impurity decrease

$$\begin{aligned}\Delta(s,\mathrm{t},r) \coloneqq\ & \pi(\mathrm{t})\mathrm{Var}\{r \mid \mathbf{x} \in \mathrm{t}\} \\ & - \pi(\mathrm{t}_L)\mathrm{Var}\{r \mid \mathbf{x} \in \mathrm{t}_L\} - \pi(\mathrm{t}_R)\mathrm{Var}\{r \mid \mathbf{x} \in \mathrm{t}_R\},\end{aligned} \quad [7]$$

instead of the finite sample impurity decrease $\hat{\Delta}(s, \mathrm{t}, \mathbf{r})$.

2. Values of the new children nodes t_L *and* t_R *are obtained by adding, to the value of* t*, the population means* $\mathbb{E}_\pi\{r \mid \mathrm{t}_L\}$ *and* $\mathbb{E}_\pi\{r \mid \mathrm{t}_R\}$ *respectively, instead of the sample means* $\bar{r}_{\mathrm{t}_L}$ *and* $\bar{r}_{\mathrm{t}_R}$ *respectively.*

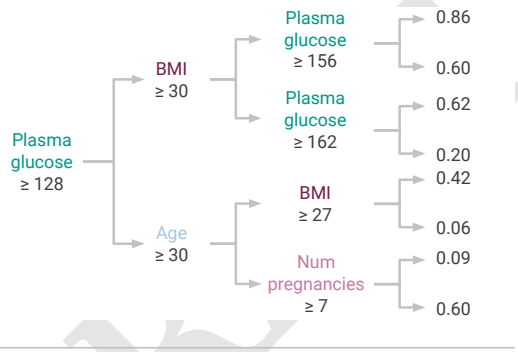
Then for each fitted tree $\hat{f}_k$*, the set of features split upon is contained within a single index set* I_k *for some* k*.*

The proof for this theorem is deferred to Appendix S6. Note that for any generative model $h(\mathbf{x}, y)$, the finite-sample impurity decrease converges to the population impurity decrease as the sample size goes to infinity: $n^{-1}\hat{\Delta}(s, \mathrm{t}, \mathbf{h}) \to \Delta(s, \mathrm{t}, h)$ and $\bar{h}_\mathrm{t} \to \mathbb{E}\{h \mid \mathrm{t}\}$ as $n \to \infty$. This justifies our claim that the modified algorithm is equivalent to running FIGS in the large sample limit. Note that the number of terms in the fitted model need not be equal to the number of additive components K.

Real-world example. We now show how disentanglement could lead to a more scientific accurate and interpretable models for a Diabetes classification dataset (43, 67). In this dataset, eight risk factors were collected and used to predict the onset of diabetes within five years. The dataset consists of 768 female subjects from the Pima Native American population near Phoenix, AZ, USA. 268 of the subjects developed diabetes, which is treated as a binary label.

Fig S1 shows two models, one learned by FIGS and one learned by CART. In both models, the prediction corresponds to the risk of diabetes onset withing five years (a higher prediction corresponds to a higher risk). Both achieve roughly the same performance (FIGS yields an AUC of 0.820 whereas CART yields an AUC of 0.817), but the models have some key differences. The FIGS model includes fewer features and fewer total splits than the CART model, making it easier to understand in its entirety. Moreover, the FIGS model has no interactions between features, making it clear that each of the features contributes independently of one another, something which any single-tree model is unable to do.

CART



FIGS

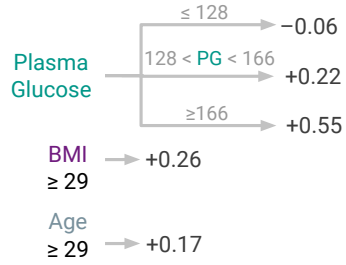


Fig. S1. Comparison between FIGS and CART on the diabetes dataset. FIGS learns a simpler model, which disentangles interactions between features. Both models achieve the same generalization performance (FIGS yields an AUC of 0.820 whereas CART yields 0.817.)

D. Proof of Theorem 1.

Proof of Theorem 1. We prove this by induction on the total number of splits, with the base case being trivial. By the induction hypothesis, we may assume WLOG that $\hat{f}_1$ only has splits on features in I_1. Consider a candidate split s on a leaf $\mathrm{t} \in \hat{f}_1$ based on a feature $m \in I_2$. Let t' denote the projection of t onto I_1. As sets in $\mathbb{R}^d$, we may then write

$$\mathrm{t} = \mathrm{t}' \times \mathbb{R}^{[d]\setminus I_1}, \qquad [8]$$

$$\mathrm{t}_L = \mathrm{t}' \times (-\infty, \tau] \times \mathbb{R}^{[d]\setminus I_1 \cup \{m\}}, \qquad [9]$$

and

$$\mathrm{t}_R = \mathrm{t}' \times (\tau, \infty) \times \mathbb{R}^{[d]\setminus I_1 \cup \{m\}}. \qquad [10]$$

Recall that we work with the residual $r = f(\mathbf{x}) - \sum_{k=1}^{K} \hat{f}_k$. Now using the law of total variance, we can rewrite the weighted impurity decrease in a more convenient form:

$$\Delta(s, \mathrm{t}, r) = \frac{\pi(\mathrm{t}_L)\pi(\mathrm{t}_R)}{\pi(\mathrm{t})}(\mathbb{E}\{r \mid \mathbf{x} \in \mathrm{t}_L\} - \mathbb{E}\{r \mid \mathbf{x} \in \mathrm{t}_R\})^2. \quad [11]$$

We may assume WLOG that this quantity is strictly positive. By the induction hypothesis, we can divide the set of component trees into two collections, one of which only splits on features in I_2, and those which only split on features in $[d]\backslash I_2$. Denoting the function associated with the second collection of trees by g_2, we observe that

$$\begin{aligned}\mathbb{E}\{r \mid \mathbf{x} \in \mathrm{t}_L\} - \mathbb{E}\{r \mid \mathbf{x} \in \mathrm{t}_R\} &= \mathbb{E}\{f - g \mid \mathbf{x} \in \mathrm{t}_L\} - \mathbb{E}\{f - g \mid \mathbf{x} \in \mathrm{t}_R\} \\ &= \mathbb{E}\{f_2 - g_2 \mid \mathbf{x} \in \mathrm{t}_L\} - \mathbb{E}\{f_2 - g_2 \mid \mathbf{x} \in \mathrm{t}_R\}.\end{aligned}$$

Since f_2 and g_2 do not depend on features in I_1, we can then further rewrite this quantity as

$$\mathbb{E}\{f_2 - g_2 \mid x_m \leq \tau\} - \mathbb{E}\{f_2 - g_2 \mid x_m > \tau\}. \quad [12]$$

Meanwhile, using Eq. (8), Eq. (9), and Eq. (10), we may rewrite

$$\frac{\pi(\mathrm{t}_L)\pi(\mathrm{t}_R)}{\pi(\mathrm{t})} = \pi_1(\mathrm{t}')\pi_2(x_m \leq \tau)\pi_2(x_m > \tau). \quad [13]$$

Plugging Eq. (12) and Eq. (13) back into Eq. (11), we get

$$\Delta(s, \mathrm{t}, r) = \pi_1(\mathrm{t}')\pi_2(x_m \leq \tau)\pi_2(x_m > \tau)(\mathbb{E}\{f_2 - g \mid x_m \leq \tau\} - \mathbb{E}\{f_2 - g \mid x_m > \tau\})^2. \quad [14]$$

In contrast, if we split a new root node t_0 on m at the same threshold and call this split s', we can run through a similar set of calculations to get

$$\Delta(s', \mathrm{t}_0, r) = \pi_2(x_m \leq \tau)\pi_2(x_m > \tau)(\mathbb{E}\{f_2 - g \mid x_m \leq \tau\} - \mathbb{E}\{f_2 - g \mid x_m > \tau\})^2. \quad [15]$$

Comparing Eq. (14) and Eq. (15), we see that

$$\Delta(s, \mathrm{t}, r^{(-1)}) = \pi_1(\mathrm{t}')\Delta(s', \mathrm{t}_0, r),$$

and as such, split s' will be chosen in favor of s. □

E. Theoretical generalization upper bounds. We shall prove a generalization upper bound showing that disentangled tree-sum models enjoy better prediction performance that single-tree models when fitted to data with additive structure. When the data is generated from a fully additive model with C^1 component functions (i.e. the generative model Eq. (6) with blocks of size $d_k = 1$, and $f_k \in C^k([0,1])$ for each k), it was recently shown that any single decision tree model has a squared error generalization lower bound of $\Omega(n^{-2/(d+2)})$. This is significantly worse than the minimax rate of $\tilde{O}\left(dn^{-2/3}\right)$ for this problem (13). Tight generalization upper bounds have proved elusive for CART due to the complexity of analyzing the tree growing procedure, and are difficult for FIGS for the same reason. Nonetheless, we can prove a partial result that shows that a disentangled tree-sum representation is more effective than a single tree model. To formalize this, let $\mathcal{C} = \{\mathcal{T}_1, \ldots, \mathcal{T}_M\}$ be the collection of tree structures learnt by FIGS. We say that a function is a *tree-sum implementable with* $\mathcal{C}$ if it can be represented as a sum of component functions, each of which is implementable by one of the tree structures $\mathcal{T}_k$ (i.e. constant on each leaf of $\mathcal{T}_k$). We use $\mathcal{F}(\mathcal{C})$ to denote the collection of all tree-sum implementable with $\mathcal{C}$. As discussed in Sec 2, FIGS essentially fits an empirical risk minimizer from $\mathcal{F}(\mathcal{C})$. In the following theorem, we prove that when $\mathcal{C}$ is instead chosen by an oracle, an empirical risk minimizer with respect to $\mathcal{F}(\mathcal{C})$ has good generalization properties.

Theorem 2 (Generalization upper bounds using oracle tree structures). *Consider the generative model* Eq. (6)*, and further suppose the distribution π_k of each independent block $\mathbf{x}_{I_k}$ has a continuous density, each f_k is C^1, with $\|\nabla f_k\|_2 \leq \beta_k$, and that ϵ is homoskedastic with variance $\mathbb{E}\left\{\epsilon^2 \mid \mathbf{x}\right\} = \sigma^2$. For any sample size n, there exists an oracle collection of trees $\mathcal{C} = \{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$ disentangling* Eq. (6) *such that for a training set $\mathcal{D}_n$ of size n, any empirical risk minimizer $\tilde{f}$ of $\mathcal{F}(\mathcal{C})$ satisfies the following squared error generalization upper bound:*

$$\mathbb{E}_{\mathcal{D}_n, \mathbf{x}}\left\{(\tilde{f}(\mathbf{x}) - f(\mathbf{x}))^2 \mathbf{1}\{\mathcal{E}^c\}\right\} \leq \sum_{k=1}^{K} c_k \left(\frac{\sigma^2}{n}\right)^{\frac{2}{d_k+2}}. \quad [16]$$

Here, $\mathcal{E}$ is an event with vanishing probability $\mathbb{P}\{\mathcal{E}\} = O(n^{-2/(d_{max}+2)})$ where $d_{max} = \max_k d_k$ is the size of the largest feature block in Eq. (6)*, while $c_k := 8\left(d_k \beta_k^2 \|\pi_k\|_\infty\right)^{\frac{d_k}{d_k+2}}$.* [§§§]

The proof of Theorem 2 can be found in Appendix D, and builds on recent work (66) which shows how to interpret decision trees as linear models on a set of engineered features corresponding to internal nodes in the tree. This interpretation has a natural extension to FIGS. It is instructive to consider two extreme cases: If $d_k = 1$ for each k, then we have an upper bound of $O\left(dn^{-2/3}\right)$. If on the other hand $K = 1$, we have an upper bound of $O\left(n^{-2/(d+2)}\right)$. Both (partially oracle) bounds match the well-known minimax rates for their respective inference problems (54).

[§§§] We note that the error event $\mathcal{E}$ is due to the query point possibly landing in leaf nodes containing very few or even zero training samples, which can be thus be detected and avoided in practice by imputing a default value.

F. Proof of Theorem 2.

Proof of Theorem 2. To construct $\mathcal{C}$, for each k, construct a tree $\mathcal{T}_k$ that partitions $[0,1]^{d_k}$ into cubes of side length h_k, where h_k is a parameter to be determined later. Let p_k denote the number of internal nodes in $\mathcal{T}_k$. Recall that we use $\tilde{f}$ to denote the empirical risk minimizer among all functions in $\mathcal{F}(\mathcal{C})$. Let g denote an ℓ_2 risk minimizer in $\mathcal{F}(\mathcal{C})$. For any event $\mathcal{E}$, we then apply Cauchy-Schwarz to get

$$\mathbb{E}_{\mathcal{D}_n,\mathbf{x}\sim\pi}\Big\{\big(\tilde{f}(\mathbf{x})-f(\mathbf{x})\big)^2\mathbf{1}\{\mathcal{E}^c\}\Big\} \le 2\mathbb{E}\big\{(f(\mathbf{x})-g(\mathbf{x}))^2\big\} + 2\mathbb{E}\Big\{\big(g(\mathbf{x})-\tilde{f}(\mathbf{x})\big)^2\mathbf{1}\{\mathcal{E}^c\}\Big\}. \tag{17}$$

The first term represents bias, and quantifies the error incurred when attempting to approximate f with the function class $\mathcal{F}(\mathcal{C})$, while the second term represents variance, and quantifies the error incurred because of sampling uncertainty. We will bound the first term using smoothness properties of the f_k's, while for the second term, we will rewrite $\tilde{f}$ and g as linear functions in an engineered feature space, which would allow us to use calculations from linear modeling theory to obtain a bound.

We begin by bounding the second term, and assume WLOG that $\mathbb{E}_{\pi_k}\{f_k\}=0$ for each k. Let Ψ_k be the feature mapping associated with $\mathcal{T}_k$ as described earlier in this section, and concatenate them for $k=1,\ldots,K$ to get the mapping Ψ. There is a one-to-one correspondence between $\mathcal{F}(\mathcal{C})$ (tree-sum functions implementable with $\mathcal{C}$) and linear function with respect to the representation Ψ (i.e. of the form $\mathbf{x}\mapsto\boldsymbol{\theta}^T\Psi(\mathbf{x})$ for some vector $\boldsymbol{\theta}$). As such, $\tilde{f}$, the empirical risk minimizer in $\mathcal{F}(\mathcal{C})$, can be written as $\tilde{f}(\mathbf{x})=\tilde{\boldsymbol{\theta}}^T\Psi(\mathbf{x})$, where $\tilde{\boldsymbol{\theta}}$ is the solution to the least squares problem

$$\min_{\boldsymbol{\theta}}\sum_{i=1}^n\big(\boldsymbol{\theta}^T\Psi(\mathbf{x}_i)-y_i\big)^2.$$

Furthermore, one can check that the ℓ_2 risk minimizer g can be written as $g(\mathbf{x})=\boldsymbol{\theta}^{*T}\Psi(\mathbf{x})$, where

$$\boldsymbol{\theta}^*(\mathrm{t}) \coloneqq \frac{\sqrt{N(\mathrm{t}_L)N(\mathrm{t}_R)}}{N(\mathrm{t})}(\mathbb{E}\{y\mid\mathrm{t}_L\}-\mathbb{E}\{y\mid\mathrm{t}_R\}) \tag{18}$$

for each node t. This gives the formula $g(\mathbf{x})=\sum_{k=1}^K g_k(\mathbf{x}_{I_k})$, where for each k,

$$g_k(\mathbf{x}_{I_k}) \coloneqq \mathbb{E}\big\{f_k(\mathbf{x}'_{I_k})\mid\mathbf{x}'_{I_k}\in\mathrm{t}_k(\mathbf{x}_{I_k})\big\}.$$

Here, $\mathrm{t}_k(\mathbf{x}_{I_k})$ is the leaf in $\mathcal{T}_k$ containing $\mathbf{x}_{I_k}$, and $\mathbf{x}'_{I_k}$ an independent copy of $\mathbf{x}_{I_k}$.

Meanwhile, note that we have the equation

$$y=\boldsymbol{\theta}^{*T}\Psi(\mathbf{x})+\eta+\epsilon$$

where $\eta\coloneqq f(\mathbf{x})-g(\mathbf{x})$ satisfies

$$\begin{aligned}\mathbb{E}\{\eta\mid\Psi(\mathbf{x})\} &= \mathbb{E}\left\{\sum_{k=1}^K(f_k(\mathbf{x}_{I_k})-g_k(\mathbf{x}_{I_k}))\mid\Psi(\mathbf{x})\right\}\\ &= \sum_{k=1}^K\mathbb{E}\{f_k(\mathbf{x}_{I_k})-g_k(\mathbf{x}_{I_k})\mid\Psi_k(\mathbf{x}_{I_k})\}\\ &= 0.\end{aligned}$$

As such, we may apply Theorem 7 with the event $\mathcal{E}$ given in the statement of the theorem to get

$$\mathbb{E}\Big\{\big(g(\mathbf{x})-\tilde{f}(\mathbf{x})\big)^2\mathbf{1}\{\mathcal{E}^c\}\Big\} \le 2\left(\frac{p\sigma^2}{n+1}+2\mathbb{E}\big\{(f(\mathbf{x})-g(\mathbf{x}))^2\big\}\right).$$

Plugging this into Eq. (17), we get

$$\mathbb{E}_{\mathcal{D}_n,\mathbf{x}\sim\pi}\Big\{\big(\tilde{f}(\mathbf{x})-f(\mathbf{x})\big)^2\mathbf{1}\{\mathcal{E}^c\}\Big\} \le 10\mathbb{E}\big\{(f(\mathbf{x})-g(\mathbf{x}))^2\big\}+\frac{4p\sigma^2}{n+1}. \tag{19}$$

By independence, and the fact that $\mathbb{E}\{g_k(\mathbf{x}_{I_k})\}=0$ for each k, we can decompose the first term as

$$\mathbb{E}\big\{(f(\mathbf{x})-g(\mathbf{x}))^2\big\}=\sum_{k=1}^K\mathbb{E}\big\{(f_k(\mathbf{x})-g_k(\mathbf{x}))^2\big\}.$$

This allows us to decompose the right hand side of Eq. (19) into the sum

$$\sum_{k=1}^K\left(10\mathbb{E}\big\{(f_k(\mathbf{x}_{I_k})-g_k(\mathbf{x}_{I_k}))^2\big\}+\frac{4p_k\sigma^2}{n+1}\right). \tag{20}$$

We reduce to the case of uniform distribution μ, via the inequality

$$\mathbb{E}_{\pi_k}\left\{(f_k(\mathbf{x}_{I_k}) - g_k(\mathbf{x}_{I_k}))^2\right\} \leq \|\pi_k\|_\infty \mathbb{E}_\mu\left\{(f_k(\mathbf{x}_{I_k}) - g_k(\mathbf{x}_{I_k}))^2\right\},$$

and from now work with this distribution, dropping the subscript for conciseness. Next, observe that

$$\mathbb{E}\left\{(f_k(\mathbf{x}_{I_k}) - g_k(\mathbf{x}_{I_k}))^2\right\} = \mathbb{E}\{\mathrm{Var}\{f_k(\mathbf{x}_{I_k}) \mid \mathfrak{t}_k(\mathbf{x}_{I_k})\}\}.$$

Using Lemma 4, we have that

$$\mathrm{Var}\{f_k(\mathbf{x}_{I_k}) \mid \mathfrak{t}_k(\mathbf{x}_{I_k})\} \leq \frac{\beta_k^2 d_k h_k^2}{6}.$$

Meanwhile, a volumetric argument gives

$$p_k \leq h_k^{-d_k}.$$

We use these to bound each term of Eq. (20) as

$$10\|\pi_k\|_\infty \mathbb{E}\left\{(f_k(\mathbf{x}_{I_k}) - g_k(\mathbf{x}_{I_k}))^2\right\} + \frac{4p_k\sigma^2}{n+1} \leq 2\|\pi_k\|_\infty \beta_k^2 d_k h_k^2 + \frac{4h_k^{-d_k}\sigma^2}{n+1}. \quad [21]$$

Pick

$$h_k = \left(\frac{2\sigma^2}{\|\pi_k\|_\infty \beta_k^2 d_k (n+1)}\right)^{\frac{1}{d_k+2}},$$

which sets both terms on the right hand side to be equal, in which case the right hand of Eq. (21) has the value

$$4\left(2\|\pi_k\|_\infty \beta_k^2 d_k\right)^{\frac{d_k}{d_k+2}} \left(\frac{\sigma^2}{n+1}\right)^{\frac{2}{d_k+2}}.$$

Summing these quantities up over all k gives the bound Eq. (16), with the error probability obtained by computing $2p/n$. □

Corollary 3. *Assume a sparse additive model, i.e. in* Eq. (6)*, assume $I_k = \{k\}$ for $k = 1, \ldots, K$. Then we have*

$$\mathbb{E}_{\mathcal{D}_n, \mathbf{x}\sim\pi}\left\{\left(\tilde{f}(\mathbf{x}) - f(\mathbf{x})\right)^2 \mathbf{1}\{\mathcal{E}^c\}\right\} \leq 8K \max_k\left(\|\pi_k\|_\infty \beta_k^2\right)^{1/3}\left(\frac{\sigma^2}{n}\right)^{\frac{2}{3}}.$$

Lemma 4 (Variance and side lengths)**.** *Let μ be the uniform measure on $[0,1]^d$. Let $\mathcal{C} \subset [0,1]^d$ be a cell. Let f be any differentiable function such that $\|\nabla f(\mathbf{x})\|_2^2 \leq \beta^2$. Then we have*

$$\mathrm{Var}_\mu\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{C}\} \leq \frac{\beta^2}{6}\sum_{j=1}^d (b_j - a_j)^2. \quad [22]$$

Proof. For any $\mathbf{x}, \mathbf{x}' \in \mathcal{C}$, we may write

$$\left(f(\mathbf{x}) - f(\mathbf{x}')\right)^2 = \left\langle \nabla f(\mathbf{x}''), \mathbf{x} - \mathbf{x}'\right\rangle^2 \leq \beta^2 \|\mathbf{x} - \mathbf{x}'\|_2^2.$$

Next, note that

$$\mathbb{E}\left\{\|\mathbf{x} - \mathbf{x}'\|_2^2 \mid \mathbf{x}, \mathbf{x}' \in \mathcal{C}\right\}^2 = \frac{1}{3}\sum_{j=1}^d (b_j - a_j)^2.$$

As such, we have

$$\begin{aligned}\mathrm{Var}_\mu\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{C}\} &= \frac{1}{2}\mathbb{E}\left\{\left(f(\mathbf{x}) - f(\mathbf{x}')\right)^2 \mid \mathbf{x}, \mathbf{x}' \in c\ell\right\} \\ &\leq \frac{\beta^2}{6}\sum_{j=1}^d (b_j - a_j)^2.\end{aligned}$$

□

G. Helper lemmas on linear regression. We consider the case of possibly under-determined least squares, i.e. the problem

$$\min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 \qquad [23]$$

where we allow for the possibility that $\mathbf{X}$ does not have linearly independent columns. When this is indeed the case, there will be multiple solutions to Eq. (23), but there is a unique element $\hat{\boldsymbol{\theta}}$ of the solution set that has minimum norm. In fact, this is given by the formula

$$\hat{\boldsymbol{\theta}} = \mathbf{X}^\dagger \mathbf{y},$$

where $\mathbf{X}^\dagger$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{X}$.

We extend the definition of leverage scores to this case by defining the i-th leverage score h_i to be the i-th diagonal entry of the matrix $\mathbf{H} \coloneqq \mathbf{X}\mathbf{X}^\dagger$. Note that the vector of predicted values is given by we have

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\theta}} = \mathbf{X}\mathbf{X}^\dagger \mathbf{y} = \mathbf{H}\mathbf{y},$$

so that this coincides with the definition of leverage scores in the linearly independent case.

In what follows, we will work extensively with leave-one-out (LOO) perturbations of the sample and the resulting estimators. We shall use $\mathbf{X}^{(-i)}$ to denote the data matrix with the i-th data point removed, and $\hat{\boldsymbol{\theta}}^{(-i)}$ to denote the solution to Eq. (23) with $\mathbf{X}$ replaced with $\mathbf{X}^{(-i)}$. We have the following two generalizations of standard formulas in the full rank setting.

Lemma 5 (Leave-one-out estimated coefficients)**.** *The LOO estimated coefficients satisfy*

$$\mathbf{x}_i^T \left(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^{(-i)} \right) = \frac{h_i \hat{e}_i}{1 - h_i}$$

where $\hat{e}_i = y_i - \mathbf{x}_i^T \hat{\boldsymbol{\theta}}$ is the residual from the full model.

Proof. Note that we may write $\mathbf{X}^\dagger = \left(\mathbf{X}^T\mathbf{X}\right)^\dagger \mathbf{X}^T$. We may then follow the proof of the usual identity but substituting Eq. (24) in lieu of the regular Sherman-Morrison formula. □

Lemma 6 (Sherman-Morrison)**.** *Let $\mathbf{X}$ be any matrix. Then*

$$\left(\mathbf{X}^{(-i)T}\mathbf{X}^{(-i)}\right)^\dagger = \left(\mathbf{X}^T\mathbf{X}\right)^\dagger + \frac{\left(\mathbf{X}^T\mathbf{X}\right)^\dagger \mathbf{x}_i \mathbf{x}_i^T \left(\mathbf{X}^T\mathbf{X}\right)^\dagger}{1 - h_i}. \qquad [24]$$

Proof. We apply Theorem 3 in (68) to $A = \mathbf{X}^T\mathbf{X}$, $c = \mathbf{x}_i$ and $d = -\mathbf{x}_i$, noting that the necessary conditions are fulfilled. □

Theorem 7 (Generalization error for linear regression)**.** *Consider a linear regression model*

$$y = \boldsymbol{\theta}^T \mathbf{x} + \epsilon + \eta$$

ϵ and η are independent, $\mathbb{E}\{\epsilon \mid \mathbf{x}\} = \mathbb{E}\{\eta \mid \mathbf{x}\} = 0$, $\mathbb{E}\{\epsilon^2 \mid \mathbf{x}\} = \sigma_\epsilon^2$, and $\mathbb{E}\{\eta^2\} = \sigma_\eta^2$. Given a training set $\mathcal{D}_n$, and a query point $\mathbf{x}$, let $\hat{\boldsymbol{\theta}}_n$ be the estimated regression vector. There is an event $\mathcal{E}$ of probability at most $2p/n$ over which we have

$$\mathbb{E}_{\mathcal{D}_n, \mathbf{x}}\left\{ \left(\mathbf{x}^T\left(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}\right)\right)^2 \mathbf{1}\{\mathcal{E}^c\} \right\} \le 2\left(\frac{p\sigma_\epsilon^2}{n+1} + 2\sigma_\eta^2 \right). \qquad [25]$$

Proof. Let $\mathcal{D}_n$ denote the training set. Let $\mathcal{E}$ be the event on which $\mathbf{x}^T\left(\mathbf{X}^T\mathbf{X}\right)^\dagger \mathbf{x} \ge \frac{1}{2}$. This quantity is the leverage score for $\mathbf{x}$, so that

$$\mathbb{E}\left\{ \mathbf{x}^T\left(\mathbf{X}^T\mathbf{X}\right)^\dagger \mathbf{x} \right\} \le \frac{p}{n+1}$$

by exchangeability of $\mathbf{x}$ with the data points in $\mathcal{D}_n$. We may then apply Markov's inequality to get

$$\mathbb{P}\{\mathcal{E}\} \le \frac{2p}{n+1}.$$

Again using exchangeability, we may write

$$\mathbb{E}_{\mathcal{D}_n, \mathbf{x}}\left\{ \left(\mathbf{x}^T\left(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}\right)\right)^2 \mathbf{1}\{\mathcal{E}^c\} \right\} = \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{E}_{\mathcal{D}_{n+1}}\left\{ \left(\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}^{(-i)} - \boldsymbol{\theta}\right)\right)^2 \mathbf{1}\{\mathcal{E}_i^c\} \right\}, \qquad [26]$$

where $\mathcal{D}_{n+1}$ is the augmentation of $\mathcal{D}_n$ with the query point $\mathbf{x}_0 = \mathbf{x}$ and response y_0, $\hat{\boldsymbol{\theta}}_{n+1}$ is the regression vector learnt from $\mathcal{D}_{n+1}$, and for each i, $\hat{\boldsymbol{\theta}}_{n+1}^{(-i)}$ that from $\mathcal{D}_{n+1} \backslash \{(\mathbf{x}_i, y_i)\}$.

To bound this, we first rewrite the prediction error for the full model as

$$\begin{aligned}\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}-\boldsymbol{\theta}\right) &= \mathbf{x}_i^T\mathbf{X}^\dagger\mathbf{y}-\mathbf{x}_i^T\boldsymbol{\theta}\\ &= \mathbf{x}_i^T\mathbf{X}^\dagger(\mathbf{X}\boldsymbol{\theta}+\boldsymbol{\epsilon}+\boldsymbol{\eta})-\mathbf{x}_i^T\boldsymbol{\theta}\\ &= \mathbf{x}_i^T\mathbf{X}^\dagger(\boldsymbol{\epsilon}+\boldsymbol{\eta}),\end{aligned} \quad [27]$$

where the last equality follows because $\mathbf{x}_i$ lies in the column space of $\mathbf{X}^\dagger\mathbf{X}$. Next, we may decompose that for the LOO model as

$$\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}^{(-i)}-\boldsymbol{\theta}\right) = \mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}^{(-i)}-\hat{\boldsymbol{\theta}}_{n+1}\right)+\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}-\boldsymbol{\theta}\right). \quad [28]$$

We expand the first term using Lemma 5 to get

$$\begin{aligned}\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}^{(-i)}-\hat{\boldsymbol{\theta}}_{n+1}\right) &= \frac{h_i}{1-h_i}\left(\mathbf{x}_i^T\hat{\boldsymbol{\theta}}_{n+1}-y_i\right)\\ &= \frac{h_i}{1-h_i}\left(\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}-\boldsymbol{\theta}\right)-\epsilon_i-\eta_i\right).\end{aligned} \quad [29]$$

We plug Eq. (29) into Eq. (28) and then Eq. (27) into the resulting equation to get

$$\begin{aligned}\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}^{(-i)}-\boldsymbol{\theta}\right) &= \frac{h_i}{1-h_i}\left(\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}-\boldsymbol{\theta}\right)-\epsilon_i-\eta_i\right)+\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}-\boldsymbol{\theta}\right)\\ &= \frac{1}{1-h_i}\left(\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}-\boldsymbol{\theta}\right)\right)-\frac{h_i}{1-h_i}(\epsilon_i+\eta_i)\\ &= \frac{1}{1-h_i}\mathbf{x}_i^T\mathbf{X}^\dagger(\boldsymbol{\epsilon}+\boldsymbol{\eta})-\frac{h_i}{1-h_i}(\epsilon_i+\eta_i).\end{aligned}$$

Taking expectations and using the independence of $\boldsymbol{\epsilon}$ and $\boldsymbol{\eta}$, we get

$$\begin{aligned}\mathbb{E}\left\{\left(\mathbf{x}_i^T\left(\hat{\boldsymbol{\theta}}_{n+1}^{(-i)}-\boldsymbol{\theta}\right)\right)^2\mathbf{1}\{\mathcal{E}_i^c\}\right\} &= \mathbb{E}\left\{\left(\frac{\mathbf{x}_i^T\mathbf{X}^\dagger\boldsymbol{\epsilon}-h_i\epsilon_i}{1-h_i}\right)^2\mathbf{1}\{\mathcal{E}_i^c\}\right\}+\mathbb{E}\left\{\left(\frac{\mathbf{x}_i^T\mathbf{X}^\dagger\boldsymbol{\eta}-h_i\eta_i}{1-h_i}\right)^2\mathbf{1}\{\mathcal{E}_i^c\}\right\}\\ &\le \mathbb{E}\left\{\left(\frac{\mathbf{x}_i^T\mathbf{X}^\dagger\boldsymbol{\epsilon}-h_i\epsilon_i}{1-h_i\wedge 1/2}\right)^2\right\}+\mathbb{E}\left\{\left(\frac{\mathbf{x}_i^T\mathbf{X}^\dagger\boldsymbol{\eta}-h_i\eta_i}{1-h_i\wedge 1/2}\right)^2\right\}.\end{aligned}$$

Summing up the first term over all indices, and then taking an inner expectation with respect to $\boldsymbol{\epsilon}$, we get

$$\begin{aligned}\frac{1}{n+1}\sum_{i=0}^{n}\mathbb{E}\left\{\left(\frac{\mathbf{x}_i^T\mathbf{X}^\dagger\boldsymbol{\epsilon}-h_i\epsilon_i}{1-h_i\wedge 1/2}\right)^2\right\} &= \frac{1}{n+1}\mathbb{E}\left\{\left\|(1-\mathrm{diag}(\mathbf{H})\wedge 1/2)^{-1}(\mathbf{H}-\mathrm{diag}(\mathbf{H}))\boldsymbol{\epsilon}\right\|_2^2\right\}\\ &= \frac{\sigma_\epsilon^2}{n+1}\mathbb{E}\left\{\mathrm{Tr}\left(\mathbf{W}\mathbf{W}^T\right)\right\}.\end{aligned} \quad [30]$$

where

$$\mathbf{W} = (1-\mathrm{diag}(\mathbf{H})\wedge 1/2)^{-1}(\mathbf{H}-\mathrm{diag}(\mathbf{H})).$$

Since $\mathbf{H}$ is idempotent, we get

$$(\mathbf{H}-\mathrm{diag}(\mathbf{H}))^2 = \mathbf{H}-\mathbf{H}\mathrm{diag}(\mathbf{H})-\mathrm{diag}(\mathbf{H})\mathbf{H}+\mathrm{diag}(\mathbf{H})^2.$$

The i-th summand in the trace therefore satisfies

$$\begin{aligned}\left(\mathbf{W}\mathbf{W}^T\right)_{ii} &= \frac{h_i-2h_i^2+h_i^2}{(1-h_i\wedge 1/2)^2}\\ &\le \frac{h_i}{1-h_i\wedge 1/2}\\ &\le 2h_i.\end{aligned}$$

Summing these up, we therefore continue Eq. (30) to get

$$\frac{\sigma_\epsilon^2}{n+1}\mathbb{E}\left\{\mathrm{Tr}\left(\mathbf{W}\mathbf{W}^T\right)\right\} \le \frac{2\sigma_\epsilon^2}{n+1}\mathbb{E}\{\mathrm{Tr}(\mathbf{H})\} \le \frac{2p\sigma_\epsilon^2}{n+1}. \quad [31]$$

Next, for any $\mathbf{x}$, we slightly abuse notation, and denote $\sigma_\eta^2(\mathbf{x}) = \mathbb{E}\left\{\eta^2 \mid \mathbf{x}\right\}$. By a similar calculation, we get

$$\frac{1}{n+1}\sum_{i=0}^{n}\mathbb{E}\left\{\left(\frac{\mathbf{x}_i^T\mathbf{X}^\dagger\boldsymbol{\eta}-h_i\eta}{1-h_i\wedge 1/2}\right)^2\right\} = \frac{1}{n+1}\mathbb{E}\left\{\mathrm{Tr}\left(\mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^T\right)\right\},$$

where $\mathbf{\Sigma}$ is a diagonal matrix with entries given by $\mathbf{\Sigma}_{ii} = \sigma_\eta^2(\mathbf{x}_i)$ for each i. We compute

$$\big(\mathbf{W}\mathbf{\Sigma}\mathbf{W}^T\big)_{ii} = \frac{\sum_{j\neq i}\mathbf{H}_{ij}^2\sigma_\eta(\mathbf{x}_j)^2}{(1-h_i\wedge 1/2)^2} \leq 4\sum_{j=0}^{n}\mathbf{H}_{ij}^2\sigma_\eta(\mathbf{x}_j)^2 = 4\big(\mathbf{H}\mathbf{\Sigma}\mathbf{H}^T\big)_{ii}.$$

This implies that

$$\begin{aligned}\frac{1}{n+1}\mathbb{E}\big\{\mathrm{Tr}\big(\mathbf{W}\mathbf{\Sigma}\mathbf{W}^T\big)\big\} &\leq \frac{4}{n+1}\mathbb{E}\big\{\mathrm{Tr}\big(\mathbf{H}\mathbf{\Sigma}\mathbf{H}^T\big)\big\}\\ &\leq \frac{4}{n+1}\mathbb{E}\{\mathrm{Tr}(\mathbf{\Sigma})\}\\ &= 4\sigma_\eta^2. \end{aligned} \quad [32]$$

Applying Eq. (31) and Eq. (32) into Eq. (26) completes the proof. □

Remark 8. *Note that while we have bounded the probability of $\mathcal{E}$ by $\frac{2p}{n}$, it could be much smaller in value. If Ψ is constructed out of a single tree, then $\mathcal{E}$ holds if and only if the test point lands in a leaf containing no training points. (13) shows that the probability of this event decays exponentially in n.*