SELECTING FEATURE INTERACTIONS FOR GENERALIZED ADDITIVE MODELS BY DISTILLING FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Identifying meaningful feature interactions is a central challenge in building accurate and interpretable models for tabular data. Generalized additive models (GAMs) have shown great success at modeling tabular data, but often rely on heuristic procedures to select interactions, potentially missing higher-order or context-dependent effects. To meet this challenge, we propose TabDistill, a method that leverages tabular foundation models and post-hoc distillation methods. Our key intuition is that tabular foundation models implicitly learn rich, adaptive feature dependencies through large-scale representation learning. Given a dataset, TabDistill first fits a tabular foundation model (TFM) to the dataset, and then applies a post-hoc interaction attribution method to extract salient feature interactions from it. We evaluate these interactions by then using them as terms in a GAM. Across tasks, we find that interactions identified by TabDistill lead to consistent improvements in downstream GAMs' predictive performance. We additionally introduce TabDistill-Proxy, a more efficient version of TabDistill that pretrains a probe on TFM attention scores. It selects interactions that roughly match the accuracy of TabDistill at one tenth of the cost. Our results suggest that TFMs can serve as effective, data-driven guides for interaction discovery, bridging high-capacity models and interpretable additive frameworks.

1 INTRODUCTION

As tabular machine learning becomes more widespread, interpretable models are increasingly essential in high-stakes applications and for scientific discovery, where understanding how inputs jointly influence predictions is often as important as predictive accuracy (Rudin, 2019; Molnar, 2019; Murdoch et al., 2019). Generalized additive models (GAMs) are a widely used class of interpretable models that decompose predictions into feature-level effects, offering transparency and strong performance on tabular data (Hastie & Tibshirani, 1987; Lou et al., 2012; Agarwal et al., 2021).

While GAMs have shown strong performance, they often struggle in real-world problems that involve feature interactions which cannot be captured solely through an additive model. Existing approaches for selecting interactions to use GAMs typically rely on greedy heuristics or model-specific criteria, such as sequentially adding pairwise terms based on residual improvements. These methods are limited in their ability to detect a small, meaningful set of interactions across diverse datasets.

Meanwhile, modern tabular foundation models (TFMs) have demonstrated an impressive capability to model rich feature interactions, achieving it through large-scale pretraining and in-context learning (Hollmann et al., 2022; 2025; Qu et al., 2025). While TFMs capture complex interactions, their large size and opaque structure make them difficult to interpret or directly deploy in settings that require interpretability.

To meet this challenge, we propose TabDistill, a method that bridges TFMs and interpretable models by using TFMs to identify interactions that are then used as additive terms in an interpretable GAM (see Figure 1). Given a dataset, TabDistill first fits a TFM. Then, a scalable post-hoc interaction attribution method is applied to the fitted TFM to identify important feature interactions. Finally, these interactions are incorporated as additive terms in a GAM. Since post-hoc attribution requires many queries to the TFM, we further propose TabDistill-Proxy, which replaces per-instance attribution with

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

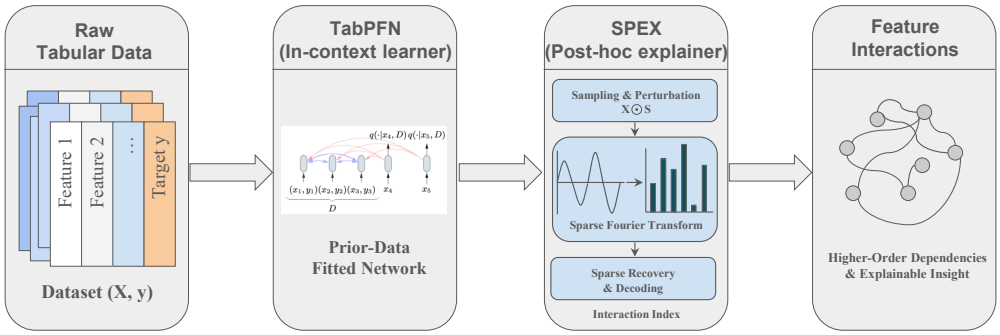


Figure 1: TabDistill utilizes the adaptive feature dependency structure learned from TFM. The distilled model is then passed through the post-hoc explainer SPEX for interaction feature discovery.

a feature-pair scoring model pretrained on TabPFN's internal attention. It selects interactions roughly an order of magnitude faster than TabDistill while matching its downstream accuracy.

We conduct experiments on TabDistill using TabPFN and post-hoc attribution methods (particularly the Faith-Banzhaf Interaction Index (Tsai et al., 2023) computed via SPEX (Kang et al., 2025)). Across a range of real-world and synthetic benchmarks, we show that interactions selected by TabDistill consistently improve the predictive performance of downstream GAMs compared to standard heuristic and post-hoc baselines, that this advantage carries over to larger datasets and to a different TFM, and that TabDistill-Proxy retains it at a fraction of the cost. These results suggest that TFMs can serve as effective, data-driven guides for interaction discovery, enabling interpretable additive models to benefit from the representational power of modern TFMs without sacrificing interpretability.

2 Related work

GAMs Generalized Additive Models (GAMs) (Hastie & Tibshirani, 1987) emerged as a generalization of Generalized Linear Models (Nelder & Wedderburn, 1972) which include non-linear transformations of the input features. Traditional GAMs often use splines and backfitting (Hastie & Tibshirani, 1987), enhanced by penalized regression splines (Wood, 2003) and fast fitting algorithms (Wood, 2001). Spline-based GAMs use the backfitting algorithm, iteratively updating each shape function to fit the residuals of others until convergence. More recent advances include Explainable Boosting Machines (EBMs) (Lou et al., 2012; 2013; Caruana et al., 2015), which use decision trees to model shape functions via cyclic gradient boosting and are widely used to model treatment effects in medical data. On the other hand, Neural Additive Models (Agarwal et al., 2021) and related works (Chang et al., 2022; Dubey et al., 2022; Radenovic et al., 2022; Xu et al., 2022; Enouen & Liu, 2022; Bouchiat et al., 2024; Yang et al., 2021) use multilayer perceptrons as non-linear transformations to model the shape functions.

Model-agnostic interaction detection We build off of the recently proposed SPEX (Kang et al., 2025). We compare against a set of post-hoc methods based on game-theoretic notions, described in Section 4 (Tsai et al., 2023). Other recent methods are model-specific, e.g. for trees (Basu et al., 2018) or for a neural network (Tsang et al., 2017).

Tabular foundation models (TFMs) TFMs adapt the in-context learning paradigm to predictive modeling for tabular data. Building on this idea, Hollmann et al. (2022) introduce TabPFN, a transformer-based model trained exclusively on synthetic tabular datasets that performs prediction via in-context learning, thereby eliminating task-specific optimization at test time. Subsequent work, TabPFN-2 (Hollmann et al., 2025), substantially improves scalability, supporting up to 10,000 samples and 500 features. More recently, TabPFN-3 (Grinsztajn et al., 2026) scales this regime to datasets with up to one million training rows and 200 features while substantially reducing training and inference time. Besides this line of work, a variety of models have been proposed that target different inference-time tradeoffs (e.g. TabICL (Qu et al., 2025)) or different task settings, e.g. causal discovery (Robertson et al., 2025; Bynum et al., 2025; Balazadeh et al., 2025; Ma et al., 2025),

Algorithm 1 Interaction Selection via TabDistill

Input: Dataset (X, y), number of interactions k
Output: Top-k interactions $\mathcal{I}_{\text{top}}$ for a GAM model
Fit TabPFN on (X, y) to obtain model $\mathcal{A}$
Initialize an empty multiset $\mathcal{C}$
for each sample (X_i, y_i) in (X, y) **do**
  $\mathcal{I}_i \leftarrow \text{SPEX}(\mathcal{A}, X_i, y_i)$
  Add all interactions in $\mathcal{I}_i$ to $\mathcal{C}$
end for
Count the frequency of each interaction in $\mathcal{C}$ and select the k most frequent interactions as $\mathcal{I}_{\text{top}}$ for downstream GAM fitting
return $\mathcal{I}_{\text{top}}$

statistical inference (Peyrard & Cho, 2025), or for directly producing an interpretable model (Mueller et al., 2024; Zhuang et al., 2024).

Distillation A variety of works have studied distilling pre-trained black-box models into interpretable models in various domains such as computer vision (Ghosh et al., 2023; Ha et al., 2021), speech (Yang et al., 2023), reinforcement learning (Dispoto et al., 2025), natural-language processing (Singh et al., 2023), and tabular data (Tan et al., 2018; Si et al., 2024). In black-box models, distillation has been found to improve the relative interpretability of the student model (Han et al., 2023; Liu et al., 2018).

3 Methods: TabDistill

3.1 Problem Setup

A GAM with interactions has the form

$$g(\mathbb{E}[y \mid x]) = \beta + \underbrace{\sum_{j=1}^{p} f_j(x_j)}_{\text{univariate shape functions}} + \underbrace{\sum_{I \in \mathcal{I}} f_I(x_I)}_{\text{interaction terms}}, \tag{1}$$

where $x \in \mathbb{R}^p$ is the input, y the response, g a link function (identity for regression, logit or softmax for classification), β an intercept, and each shape function $f_j : \mathbb{R} \to \mathbb{R}$ captures the effect of one feature and can be inspected directly. Each interaction term f_I depends jointly on a set I of two or more features and is non-additive in the sense of Tsang et al. (2017). Interactions can improve accuracy but cost interpretability, so we want as few of them as possible.

Following Lou et al. (2013), we write $\mathcal{U}_K$ for the set of all feature subsets of size at most K, so that every interaction term in Equation (1) is indexed by some $I \in \mathcal{U}_K$. For a chosen interaction set $\mathcal{I} \subseteq \mathcal{U}_K$, let $f_{\mathcal{I}}$ denote the GAM of Equation (1) fitted with exactly these interaction terms. Our goal is to find a small set $\mathcal{I}$ that minimizes the expected loss of the resulting model,

$$\min_{\mathcal{I} \subseteq \mathcal{U}_K,\, |\mathcal{I}| \le N_{int}} \mathbb{E}\big[L\big(y, f_{\mathcal{I}}(x)\big)\big],$$

where L is a non-negative loss function and N_{int} is the number of interaction terms. Since the number of candidate sets grows combinatorially in p and K, we do not solve this problem exactly; instead, TabDistill produces a ranked list of candidate interactions.

3.2 TabDistill Algorithm

TabDistill takes in a supervised dataset and outputs a ranked list of important feature interactions for use in the downstream GAM. TabDistill leverages the intuition that tabular foundation models implicitly learn rich, adaptive feature dependencies through large-scale representation learning.

As its first step, TabDistill fits a TFM to the dataset. Then, TabDistill applies a post-hoc interaction attribution method to extract salient feature interactions from the fitted model. While many choices can be made for the post-hoc attribution method, we desire a method that does not require internal access to the TFM (so it is compatible with a variety of architectures, including those not yet released),

is fairly sample-efficient (to avoid making too many computationally intensive calls to the TFM), and can effectively find predictive interactions. To meet these criteria, we use the Faith-Banzhaf Interaction Index (FBII) (Tsai et al., 2023), calculated via SPEX (Kang et al., 2025). SPEX learns an approximate sparse Fourier transform surrogate $\hat{f}(x) \approx f(x) = \sum_{u \in \mathcal{U}_K} f_u(x_u)$. It efficiently finds a small set of $\mathbf{k}$ with $|\mathbf{k}| \ll n$, denoted $\mathcal{K}$, and $\hat{F}(\mathbf{k})$ for each $\mathbf{k} \in \mathcal{K}$ such that

$$\hat{f}(\mathbf{m}) = \sum_{\mathbf{k} \in \mathcal{K}} (-1)^{\langle \mathbf{m}, \mathbf{k} \rangle} \hat{F}(\mathbf{k}),$$

where $\mathbf{m} \in \mathbb{F}_2^n$ is a binary masking vector. SPEX recovers sparse, low-order Fourier coefficients corresponding to feature interactions, from which we compute the FBII as post-processing. By combining sparse Fourier analysis with structured masking and efficient decoding, SPEX avoids enumerating the exponential interaction space, enabling faithful interaction-based explanations in long-context settings beyond marginal effects.

With $\hat{f}$ and $\hat{F}$ identified by SPEX, TabDistill then identifies the feature importance coefficients in the surrogate function and can specify the important interaction effects in the distilled foundation model. Algorithm 1 summarizes the interaction search procedure of TabDistill.

3.3 TabDistill-Proxy: Efficient Pairwise Interaction Selection

TabDistill explains every training instance with SPEX, and each explanation requires many masked TabPFN evaluations, so its cost grows with both the number of instances and the SPEX query budget. TabDistill-Proxy removes this cost by giving up query-only access: it reads the TFM's internal feature attention once per dataset and scores feature pairs with a *Global Learner* pretrained on separate calibration datasets. The current version selects pairwise interactions for binary classification.

Feature-pair evidence from attention. We fit an additive GAM, take its out-of-fold residuals r (what an interaction-free model cannot explain), fit TabPFN with r as the regression target, and extract its feature attention, aggregated over rows with weights that emphasize high-residual rows. Each layer/head is a *unit*, U in total. Because TabPFN-2 packs several features into one token, attention is measured between feature groups, so we repeat the measurement under random feature permutations that place each pair in different groups and estimate each pair's contribution per unit from the group-level attention, repeating the estimation on subsets of permutations. A pair's score in a unit is the proportion of fits with a nonzero coefficient times the median absolute value of those coefficients; ranking scores within each unit and normalizing to $[0, 1]$ gives an evidence matrix $E \in [0, 1]^{P \times U}$ over the P feature pairs.

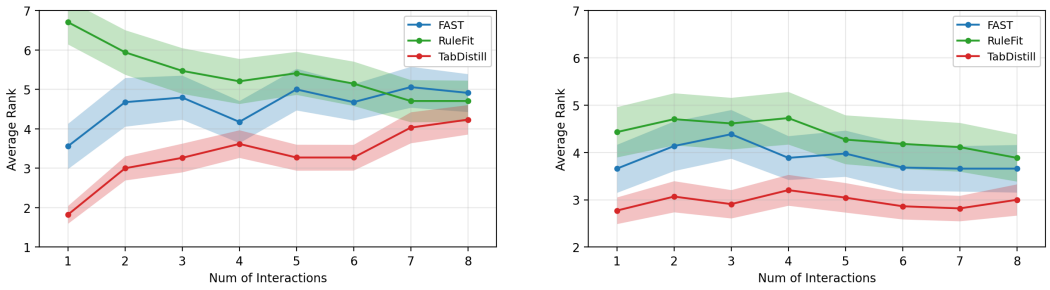
Pretraining the Global Learner. On each calibration dataset we compute E, shortlist candidate pairs, and label each by the validation gain of a GAM with that pair as its only interaction over an additive GAM: the five best candidates with positive gain receive labels decreasing with rank, all others zero. Pooling evidence vectors and labels across calibration datasets, we fit an ElasticNet with nonnegative coefficients, which yields one weight per unit, $\mathbf{w} \in \mathbb{R}^U_{\geq 0}$.

Scoring and selecting pairs on a target dataset. Each pair p of a target dataset receives a learned score $e_p = E_{p,:}\mathbf{w}$, a consensus score $c_p = \left(\frac{1}{U}\sum_{u=1}^{U} E_{p,u}^2\right)^{1/2}$ that treats all units equally, and a blend of their within-dataset percentile ranks, $b_p = (1-\gamma)\,\mathrm{rankpct}(c_p) + \gamma\,\mathrm{rankpct}(e_p)$ with a fixed $\gamma \in [0, 1]$. The top-ranked pairs under the three scores form a candidate pool, and for a requested number n of interactions the default choice is the top-n pairs under b_p. Besides the default set, we obtain two alternative candidates by screening all size-n subsets of the pool, and select the best of the three on an independent validation split. Full details are in Section A.1.

4 Results

4.1 Experimental setup

Datasets We select datasets in TabArena (Erickson et al., 2025), TALENT (Ye et al., 2024), and PMLB (Olson et al., 2017) that have fewer than 2,000 samples and fewer than 10 features. After removing deprecated datasets in PMLB, we included 35 regression and 44 classification task datasets.



(a) Regression tasks (MAE). (b) Classification tasks (F1).

Figure 2: *Comparison of TabDistill against baseline methods*, measured by the average rank across datasets (lower is better). TabDistill ranks best across settings; the gap is largest at small interaction budgets, where choosing informative interactions matters most, and narrows as more are added.

Baselines As a baseline, we run FAST (Lou et al., 2013), the standard greedy interaction selection heuristic in the InterpretML package (Lou et al., 2012; Caruana et al., 2015; Nori et al., 2019). As a second baseline, we compare against RuleFit (Friedman & Popescu, 2008), which selects rules from a random forest and ranks them based on their support (using the default hyperparameters implemented in the imodels package (Singh et al., 2021)).

For TabDistill, we choose TabPFN-2 as the TFM to be distilled and SPEX with the feature interaction index FBII as the post-hoc explanation method. Since the computational cost of the post-hoc inference is high, we use a sample budget $B = 500$ in SPEX. We also compare against different variations of post-hoc interaction detection methods, including the Faith-Shapley Interaction Index (FSII) (Tsai et al., 2023), the Shapley Taylor Interaction Index (STII) (Sundararajan et al., 2020), the Banzhaf Interaction Index (BII) (Grabisch et al., 2000), the Shapley Interaction Index (SII) (Grabisch et al., 2000), Möbius Interactions (Mobius) (Harsanyi, 1982), and Fourier interactions (Fourier) (Ahmed et al., 1976). Unless otherwise specified, TabDistill refers to TabDistill with the FBII index.

Evaluation We use an 80/20 train–test split (seed 42); interactions are selected and EBMs are fitted on the training data, and metrics are reported on the test data. Evaluation is done by passing the resulting interactions to EBM (Lou et al., 2012), run through the InterpretML package (Nori et al., 2019) using `outer_bags` $= 4$ and `max_bins` $= 256$; all other hyperparameters are set to default. We compare TabDistill with baseline methods at each $N_{int} \in \{1, \ldots, 8\}$, using three regression metrics (MSE, MAE, and R^2) and three classification metrics (Accuracy, F1 score, and AUROC). For each metric and number of interactions, we rank nine methods, comprising two baselines (FAST and RuleFit) and seven TabDistill variants with different post-hoc feature interaction detection methods.

4.2 Main result: TabDistill identifies interactions that improve prediction performance

Comparison to interaction selection baselines. Figure 2 reports the average ranks across all datasets, where TabDistill achieves better performance than the baseline methods. A more detailed comparison is included in Appendix A.2. Beyond predictive performance, TabDistill also offers several advantages over FAST and RuleFit in terms of interaction modeling.

While FAST is computationally efficient for interaction selection, it is restricted to identifying pairwise interactions. In contrast, TabDistill is able to discover higher-order interactions, enabling it to capture more complex dependency structures that are not accessible to FAST.

As a tree-based approach, RuleFit can implicitly capture higher-order interactions. However, our results show that the interactions identified by RuleFit do not translate into comparable predictive improvements. In contrast, TabDistill selects higher-order interactions that more effectively improve downstream predictive performance.

Comparison across interpretable and black-box models. Table A4 in the appendix summarizes the average rank of different models across PMLB regression datasets. Within the interpretable models (top section), EBM consistently achieves comparable performance across this dataset collection. We

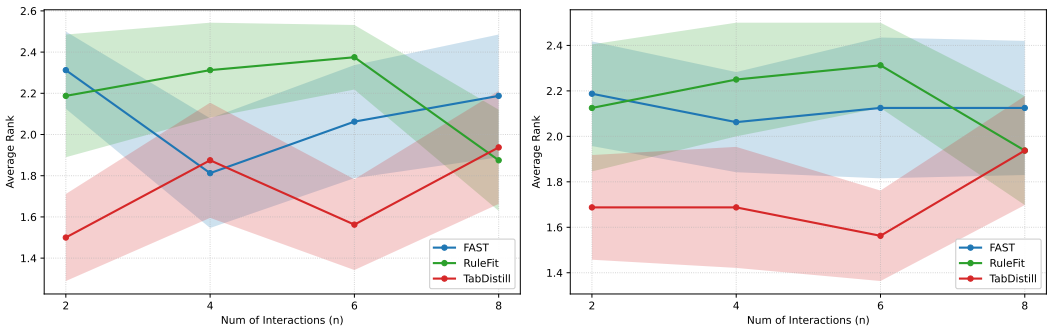


Figure 3: *TabDistill on eight larger TabArena classification datasets.* Average rank across datasets (lower is better) for F1 (left) and accuracy (right), with shaded bands of ± 1 standard error.

also point out that PyGAM (Servén & Brummitt, 2018), despite using the same TabDistill-selected interactions, performs worse than EBM. Overall, TabDistill helps reduce the performance gap between interpretable and black-box models (bottom section).

4.3 TabDistill Ablations

FBII outperforms other indices for distillation. Continuing the analysis from Section 4.2, the FBII index achieves a low average rank while also exhibiting a small variance across all post-hoc interaction indices (Table 1 (left)), indicating both strong predictive performance and high stability across different interaction budgets. These results suggest that FBII provides a particularly reliable choice of index for interaction selection in TabDistill. Consequently, unless otherwise specified, we adopt the FBII index in subsequent experiments when comparing TabDistill against baseline methods.

TabDistill works well with a limited SPEX query budget. We evaluate the effect of the sample size used for interaction selection on TabArena regression datasets. For each method, we vary the number of samples used to identify interactions from 100 to 500 and compare the top-8 selected interactions against the reference set obtained by the same method using 500 samples. Table 1 (right) shows that the interaction sets identified by TabDistill under smaller query budgets largely coincide with those obtained with 500 samples: already at 100 samples, 86% of the top-8 interactions are shared, compared with 58% for FAST and 38% for RuleFit. Consistent with this, varying the number of SPEX query samples from 100 to 500 leads to only minor changes in downstream predictive performance, with average rank variations typically within about 0.5 across interaction indices. The corresponding performance comparison is shown in Appendix Figure A3.

TabDistill scales to larger datasets. Our main experiments focus on small datasets with fewer than 2,000 samples and fewer than 10 features. To test whether TabDistill remains effective beyond this regime, we additionally evaluate it on eight TabArena classification datasets with between 10 and 100 features and fewer than 4,000 samples. We use the same protocol as before, except that SPEX uses a sample budget of $B = 1{,}000$ to accommodate the larger datasets. Figure 3 reports the average rank of TabDistill, FAST, and RuleFit across the eight datasets. Averaged over budgets, TabDistill achieves the lowest mean rank for both F1 (1.72, versus 2.09 for FAST and 2.19 for RuleFit) and accuracy (1.72, versus 2.13 for FAST and 2.16 for RuleFit). The advantage of TabDistill therefore carries over to larger, higher-dimensional data, with a smaller margin than on the small datasets.

TabDistill works with different TFMs. In order to evaluate the generality of TabDistill, we also implemented TabDistill on another TFM, TabICL (Qu et al., 2025), and evaluated it on PMLB classification datasets. Figure 4 shows that TabDistill remains effective beyond the original backbone: across different interaction budgets, TabDistill consistently achieves the best or near-best average rank compared with FAST and RuleFit. These results suggest that the benefit of TabDistill is not tied to a specific TFM and transfers well to other TFMs.

Table 1: *Performance and stability of GAMs with different interaction selection methods.* Left: mean and variance of the rank across datasets and interaction budgets N_{int}. Right: proportion of the top-8 interactions selected with 100 to 400 samples that are shared with the top-8 selected by the same method using 500 samples (higher is better), averaged over TabArena regression datasets.

Method	MAE rank		$F1$ rank		Overlap with 500-sample selection			
	Mean	Var.	Mean	Var.	100	200	300	400
TabDistill	**3.31**	0.54	2.96	0.02	0.86	0.90	0.96	0.93
FAST	4.61	0.25	3.88	0.07	0.58	0.55	0.59	0.60
RuleFit	5.41	0.44	4.37	0.09	0.38	0.50	0.54	0.51
BII	4.40	1.93	3.00	0.18	0.85	0.94	0.97	0.96
Fourier	4.27	1.74	3.37	0.07	0.85	0.94	0.97	0.95
FSII	3.44	0.71	3.13	0.04	0.86	0.90	0.91	0.93
Mobius	3.51	0.93	3.02	0.07	0.89	0.94	0.96	0.94
SII	4.04	1.51	3.12	0.14	0.85	0.93	0.95	0.93
STII	3.41	0.70	**2.92**	0.04	0.90	0.95	0.96	0.94

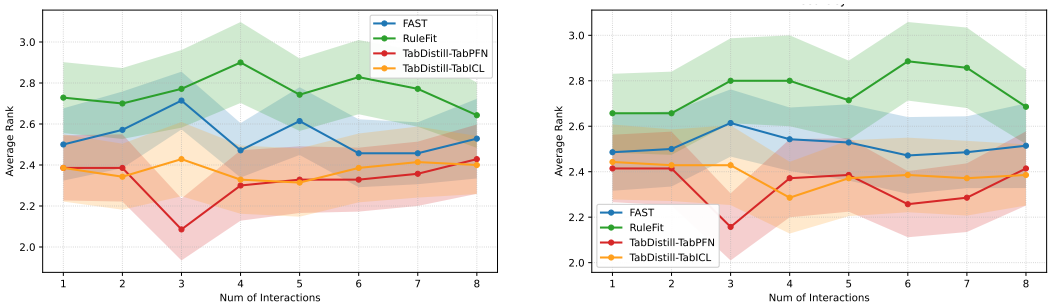


Figure 4: *TabDistill on TabICL and TabPFN against baseline methods.* Average rank across datasets, by F1 (left) and accuracy (right). Lower is better.

4.4 TabDistill-Proxy matches TabDistill at a fraction of the cost

We evaluate TabDistill-Proxy (Section 3.3) on the 26 binary classification datasets of our PMLB benchmark, with the same EBM configuration as above and N_{int} ranging from 1 to 8. The Global Learner is trained once on 30 TabArena binary classification datasets that are disjoint from PMLB, with F1 score as both the calibration gain metric and the downstream selection objective. Since RuleFit performs worse than FAST in Figure 2, we compare against the FAST baseline only.

TabDistill-Proxy matches TabDistill and outperforms FAST. Figure 5 reports the relative improvement of TabDistill-Proxy and TabDistill over FAST, averaged over datasets with one standard error. Both methods improve on FAST, and their curves nearly coincide. TabDistill-Proxy reaches this level while selecting only pairwise interactions, whereas TabDistill also has access to higher-order terms.

An order of magnitude faster than TabDistill. Figure 5(c) compares end-to-end runtime at $N_{int} = 8$. Averaged over datasets, TabDistill takes 597 s, whereas TabDistill-Proxy takes 71 s, an $8.4\times$ reduction. FAST remains the cheapest at 3 s on average. TabDistill-Proxy occupies the middle ground: it keeps most of the accuracy benefit of distilling a TFM while removing the per-instance SPEX query cost.

4.5 Analysis: TabPFN captures interaction structure in synthetic domains

To understand why TabDistill is effective, we ask a more basic question: can TabPFN reliably recover non-additive interaction structure from tabular data? We study this on Fourier-sparse functions, which directly match the structural assumption exploited by SPEX. To check that this result is not specific to Fourier-sparse functions, Section A.4 repeats the analysis with tree-structured teacher rules, whose non-smooth interaction patterns lie outside the Fourier-sparse setting. In both settings, TabPFN

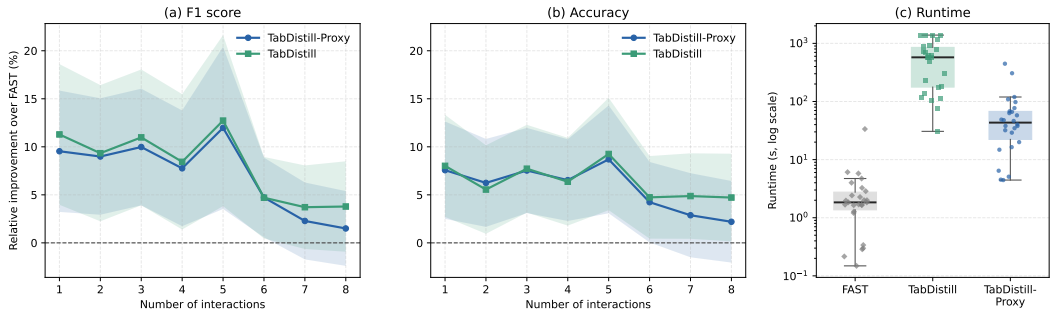


Figure 5: *TabDistill-Proxy versus TabDistill and FAST on 26 PMLB binary classification datasets.* (a), (b): relative improvement over FAST, (method − FAST)/FAST in percent, in F1 score and accuracy, averaged over datasets with shaded bands of ± 1 standard error. (c): total runtime at $N_{int} = 8$ per dataset (log scale).

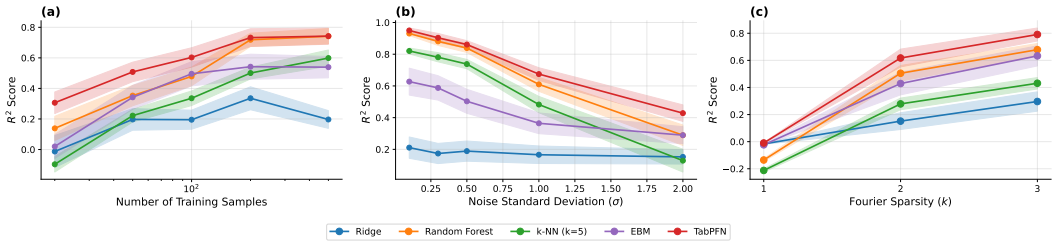


Figure 6: *Performance comparison of TabPFN against baseline methods on Fourier-sparse functions.* **(a) Low-data regime; (b) Noise robustness; (c) Extreme sparsity.**

captures the underlying interaction structure more accurately than baseline methods, supporting its role as a strong teacher model for TabDistill.

We focus on Fourier-sparse functions, since SPEX is motivated by the observation that explanation value functions are often sparse in the Boolean Fourier basis. We generate binary inputs and construct targets using a sparse Fourier expansion with low-order subsets, then evaluate TabPFN under low-data, noisy, and extreme-sparsity regimes. These experiments test whether TabPFN can recover a small number of relevant interactions from limited samples in a large combinatorial space. As shown in Figure 6, TabPFN performs strongly across all three regimes, especially in the extreme-sparsity setting where only a few active terms must be identified among exponentially many candidates.

Dataset Generation: Binary input features $\mathbf{x} \in \{0,1\}^n$ were sampled uniformly at random (i.e., each feature was an independent Bernoulli(0.5) random variable). Target functions were generated using the Walsh-Hadamard (Fourier) basis. A function is k-sparse if it can be expressed as

$$y = \sum_{i=1}^{k} c_i \chi_{S_i}(\mathbf{x}) + \varepsilon, \tag{2}$$

where $\chi_S(\mathbf{x}) = (-1)^{\sum_{j \in S} x_j}$ are basis functions indexed by feature subsets S_i, coefficients $c_i \sim \mathcal{N}(0, 2^2)$, and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise. For each function, we randomly selected k subsets S_i (always including the empty set for the constant term) with interaction order $|S_i| \leq 3$.

Experiment 1 – Low-Data Regime: With $n = 10$ features, $k = 3$ Fourier sparsity, and $\sigma = 0.5$, we varied training set size from 20 to 500 samples while holding the test set at 200 samples. This tests few-shot learning capability.

Experiment 2 – Noise Robustness: With $n = 8$ features, $k = 3$ sparsity, 300 training and 200 test samples, we varied noise standard deviation $\sigma \in \{0.1, 0.3, 0.5, 1.0, 2.0\}$ to assess robustness to different signal-to-noise ratios.

Experiment 3 – Extreme Sparsity: With $n = 15$ features ($2^{15} = 32{,}768$ possible Fourier terms), we tested $k \in \{1, 2, 3\}$ with 400 training and 200 test samples and $\sigma = 0.3$. This "needle in a haystack" setting evaluates the ability to identify minimal feature interactions in high dimensions.

Baselines: We compared TabPFN against standard ML models using their default hyperparameters: Ridge regression (with 5-fold cross-validation to select the hyperparameter $\alpha \in \{0.01, 0.1, 1.0, 10.0, 100.0\}$), Random Forest (100 trees, max depth 10, adjusted for small sample sizes), k-Nearest Neighbors ($k = 5$ with distance weighting, adjusted for small sample sizes), and EBM using default hyperparameters. Performance was measured using the R^2 score.

4.6 Case study: Predicting used Fiat 500 car price

We conduct a regression task on a used-car price prediction dataset from the TabArena benchmark (*Another Dataset on used Fiat 500*, OpenML task 363615), with 1,538 samples, seven predictor variables describing vehicle characteristics and seller location, and selling price as the target. For each of FAST, RuleFit, and TabDistill, we fit an EBM with the top 10 interactions selected by that method and evaluate on the test set. TabDistill achieves the best performance among these methods.

In terms of interpretability, TabDistill identifies a higher-order interaction among longitude, latitude, and model type shown in Figure 7. The selling price patterns in sport and non-sport models are different in the geographic space that is partitioned into 2×2 cells using the median longitude and latitude. In contrast, FAST is limited to pairwise interactions and therefore cannot directly capture the joint effect of two-dimensional location and model type, and RuleFit fails to recover this interaction. Dataset details, the EBM configuration, the full performance table, and the top 10 interactions identified by each method are provided in Appendix A.5.

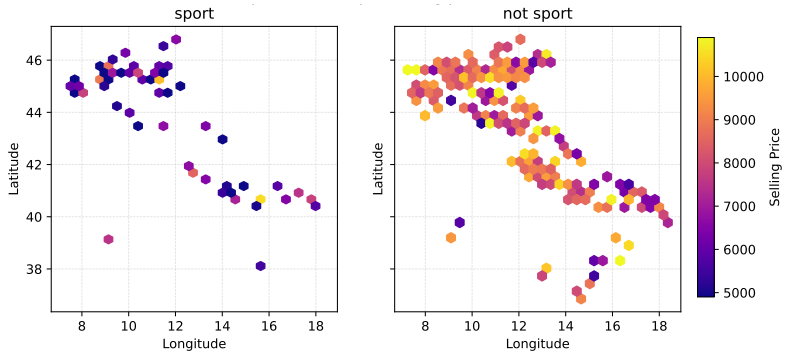


Figure 7: *A higher-order interaction identified by TabDistill between seller location and model type.* The spatial pattern differs across model types, which indicates a non-additive interaction among longitude, latitude, and model type.

5 Discussion

Overall, TabDistill provides a simple and modular way to transfer interaction structure from high-capacity tabular models to interpretable additive models, offering a practical path toward models that are both accurate and interpretable. Its utility may also extend beyond GAMs to settings where interactions themselves are scientifically meaningful (Basu et al., 2018). Additionally, these interactions may be incorporated using domain knowledge in a human-in-the-loop workflow, e.g., (Wang et al., 2021; Feng et al., 2026).

Limitations While effective, TabDistill inherits biases and failure modes of the underlying TFM, so spurious or unstable interactions may be transferred, and per-instance attribution is computationally expensive on large datasets. TabDistill-Proxy addresses the cost for pairwise interactions in binary classification, but it requires access to the TFM's attention and a pretrained Global Learner; extending it to higher-order interactions, regression, and other TFMs is a natural next step, as is training TFMs to output important interactions directly rather than relying on post-hoc distillation.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology or experiments and to implement methods. We have not used generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, assist with translation, clean and reformat the dataset, support qualitative and thematic data analysis, or interpret results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools to create figures, software code, draft parts of the research paper, and edit the research paper to improve readability. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural Additive Models: Interpretable Machine Learning With Neural Nets. In *Advances in Neural Information Processing Systems 34*, 2021.

N Ahmed, KR Rao, Ruth Bars, and Laszlo Konya. Orthogonal transform for digital signal processing. *IEEE Transactions on Systems, Man, and Cybernetics*, (11):796–797, 1976.

Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Benson Li, Junwei Ma, Jesse C Cresswell, and Rahul G Krishnan. Causalpfn: Amortized causal effect estimation via in-context learning. *arXiv preprint arXiv:2506.07918*, 2025.

Sumanta Basu, Karl Kumbier, James B. Brown, and Bin Yu. Iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences*, 115 (8):1943–1948, 2 2018. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1711236115. publisher: National Academy of Sciences section: Biological Sciences PMID: 29351989.

K. Bouchiat, A. Immer, H. YÃšche, G. Rätsch, and V. Fortuin. Improving neural additive models with bayesian principles. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4416–4443. PMLR, July 2024. URL https://proceedings.mlr.press/v235/bouchiat24a.html.

Lucius EJ Bynum, Aahlad Manas Puli, Diego Herrero-Quevedo, Nhi Nguyen, Carlos Fernandez-Granda, Kyunghyun Cho, and Rajesh Ranganath. Black box causal inference: Effect estimation via meta prediction. *arXiv preprint arXiv:2503.05985*, 2025.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 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*, pp. 1721–1730, 2015.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. Node-gam: Neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations*, 2022.

Giovanni Dispoto, Paolo Bonetti, and Marcello Restelli. " so, tell me about your policy...": Distillation of interpretable policies from deep reinforcement learning agents. *arXiv preprint arXiv:2507.07848*, 2025.

Abhimanyu Dubey, Filip Radenovic, and Dhruv Mahajan. Scalable Interpretability via Polynomials. In *Advances in Neural Information Processing Systems 35*, 2022.

James Enouen and Yan Liu. Sparse interaction additive networks via feature interaction detection and sparse selection. In *Advances in Neural Information Processing Systems 35*, 2022.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *arXiv preprint arXiv:2506.16791*, 2025.

Jean Feng, Avni Kothari, Patrick Vossler, Andrew Bishara, Lucas Zier, Newton Addo, Aaron Kornblith, Yan Shuo Tan, and Chandan Singh. Human-ai co-design for clinical prediction models. *arXiv preprint arXiv:2601.09072*, 2026.

J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008. doi: 10.1214/07-aoas148.

Shantanu Ghosh, Ke Yu, and Kayhan Batmanghelich. Distilling blackbox to interpretable models for efficient transfer learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 628–638. Springer, 2023.

Michel Grabisch, Jean-Luc Marichal, and Marc Roubens. Equivalent representations of set functions. *Mathematics of Operations Research*, 25(2):157–178, 2000.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Wooseok Ha, Chandan Singh, Francois Lanusse, Srigokul Upadhyayula, and Bin Yu. Adaptive wavelet distillation from neural networks through interpretations. *Advances in Neural Information Processing Systems*, 34:20669–20682, 2021.

Hyeongrok Han, Siwon Kim, Hyun-Soo Choi, and Sungroh Yoon. On the impact of knowledge distillation for model interpretability. *arXiv preprint arXiv:2305.15734*, 2023.

John C Harsanyi. A simplified bargaining model for the n-person cooperative game. In *Papers in game theory*, pp. 44–70. Springer, 1982.

Trevor Hastie and Robert Tibshirani. Generalized Additive Models: Some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Justin Singh Kang, Landon Butler, Abhineet Agarwal, Yigit Efe Erginbas, Ramtin Pedarsani, Bin Yu, and Kannan Ramchandran. Spex: Scaling feature interaction explanations for llms. In *Forty-second International Conference on Machine Learning*, 2025.

Xuan Liu, Xiaoguang Wang, and Stan Matwin. Improving the interpretability of deep neural networks with knowledge distillation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 905–912. IEEE, 2018.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible Models for Classification and Regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 150–158, 2012.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 623–631, 2013.

Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation models for causal inference via prior-data fitted networks. *arXiv preprint arXiv:2506.10914*, 2025.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010. doi: 10.1111/j.1467-9868.2010.00740.x.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2019.

Andreas Mueller, Julien Siems, Harsha Nori, David Salinas, Arber Zela, Rich Caruana, and Frank Hutter. Gamformer: In-context learning for generalized additive models. *arXiv preprint arXiv:2410.04560*, 2024.

W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

John Ashworth Nelder and Robert WM Wedderburn. Generalized Linear Models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. InterpretML: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(36):1–13, Dec 2017. ISSN 1756-0381. doi: 10.1186/s13040-017-0154-4. URL https://doi.org/10.1186/s13040-017-0154-4.

Maxime Peyrard and Kyunghyun Cho. Meta-statistical learning: Supervised learning of statistical inference. *arXiv preprint arXiv:2502.12088*, 2025.

Jingang Qu, David HolzmÃžller, GaÃl Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. Neural Basis Models for Interpretability. In *Advances in Neural Information Processing Systems 35*, 2022.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-pfn: In-context learning for causal effect estimation. *arXiv preprint arXiv:2506.06039*, 2025.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Daniel Servén and Charlie Brummitt. pyGAM: Generalized Additive Models in Python. URL: https://zenodo.org/records/1476122, 2018.

Jacob Si, Wendy Yusi Cheng, Michael Cooper, and Rahul G Krishnan. Interpretabnet: Distilling predictive signals from tabular data by salient feature interpretation. *arXiv preprint arXiv:2406.00426*, 2024.

Chandan Singh, Keyan Nasseri, Yan Shuo Tan, Tiffany Tang, and Bin Yu. imodels: a python package for fitting interpretable models. *Journal of Open Source Software*, 6(61):3192, 2021.

Chandan Singh, Armin Askari, Rich Caruana, and Jianfeng Gao. Augmenting interpretable models with large language models during training. *Nature Communications*, 14(1):7913, 2023.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International conference on machine learning*, pp. 9259–9268. PMLR, 2020.

Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-Compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 303–310, 2018.

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: The faithful shapley interaction index. *Journal of Machine Learning Research*, 24(94):1–42, 2023.

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

Zijie J Wang, Alex Kale, Harsha Nori, Peter Stella, Mark Nunnally, Duen Horng Chau, Mihaela Vorvoreanu, Jennifer Wortman Vaughan, and Rich Caruana. Gam changer: Editing generalized additive models with interactive visualization. *arXiv preprint arXiv:2112.03245*, 2021.

Simon N Wood. MGCV: GAMs and generalized ridge regression for R. *R news*, 1(2):20–25, 2001.

Simon N Wood. Thin plate regression splines. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(1):95–114, 2003.

Shiyun Xu, Zhiqi Bu, Pratik Chaudhari, and Ian J Barnett. Sparse neural additive model: Interpretable deep learning with feature selection via group sparsity. In *ICLR 2022 Workshop on PAIR 2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022.

Xiaoyu Yang, Qiujia Li, Chao Zhang, and Philip C Woodland. Knowledge distillation from multiple foundation models for end-to-end speech recognition. *arXiv preprint arXiv:2303.10917*, 2023.

Zebin Yang, Aijun Zhang, and Agus Sudjianto. Gami-net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition*, 120:108192, 2021.

Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and De-Chuan Zhan. A closer look at deep learning methods on tabular datasets. *arXiv preprint arXiv:2407.00956*, 2024.

Yufan Zhuang, Liyuan Liu, Chandan Singh, Jingbo Shang, and Jianfeng Gao. Learning a decision tree algorithm with transformers. *arXiv preprint arXiv:2402.03774*, 2024.

A Appendix

This appendix provides the implementation details of TabDistill-Proxy, additional benchmark results, ablation studies, an additional synthetic analysis with tree-structured interactions, and details for the Fiat 500 case study.

A.1 Implementation Details of TabDistill-Proxy

This section gives the exact settings of TabDistill-Proxy behind the description in Section 3.3 and the results in Section 4.4. Algorithms A1 to A3 give pseudocode for its three stages: computing the evidence matrix, pretraining the Global Learner, and selecting interactions on a target dataset. Table A1 collects the hyperparameters.

Attention measurements. We use the TabPFN-2 regressor with a single estimator and its input-side preprocessing disabled (no feature shuffling or rotation, no fingerprint or polynomial features, no SVD), so that the model's feature tokens correspond one-to-one to the columns we pass in. TabPFN-2 provides grouped feature attention: every token holds two raw features, so attention is measured between groups of two features rather than between individual features. Datasets with fewer than five features are padded with independent Gaussian dummy columns of scale 10^{-6} so that at least three groups exist; pairs involving dummy columns are discarded afterwards, and duplicate columns are dropped beforehand. Attention is read from layers 6–11 (zero-based, the second half of the 12-layer encoder) and all 6 heads, giving $U = 36$ layer/head units.

The out-of-fold residuals are $r_i = y_i - \hat{p}(y_i{=}1 \mid x_i)$, where $\hat{p}$ is the five-fold out-of-fold prediction of an additive EBM fitted on the training set with stratified folds. Because TabPFN predicts query rows from a labeled context, we never record attention for a row while its own residual is visible to the model. Under each feature permutation, the training rows are therefore split into two stratified halves: one half serves as the context, with r as the regression target, and feature attention is recorded for the other half as query rows; the roles are then swapped, and the two attention matrices are averaged with weights proportional to the number of query rows. Query rows are aggregated with class-balanced residual-contrast weights: in each class, rows whose $|r|$ lies in the top 30% receive weight $+0.5/n_{\text{high}}$ and the remaining rows $-0.5/n_{\text{low}}$, so the weights sum to zero and attention shared by easy and hard rows cancels. For each layer/head, the resulting group-by-group attention matrix is symmetrized as $\frac{1}{2}(A + A^\top)$ and double-centered over its off-diagonal entries; its upper triangle is the group-level measurement vector for that layer/head.

Each dataset is measured under $M = 60$ distinct feature permutations. Permutations are generated greedily so that which features share a group, which group each feature lands in, and its slot within the group are all balanced.

Evidence matrix (Algorithm A1). To estimate each original feature pair's contribution from the group-level measurement vectors, we regress them on the feature groupings across permutations. For a permutation with groups $\{g_a\}$, the regressor row for group pair (a, b) assigns weight $1/(|g_a||g_b|)$ to every feature pair (i, j) with $i \in g_a$, $j \in g_b$, and is double-centered in the same way as the measurement vectors. Stacking all permutations gives one column per feature pair and one response column per layer/head. We fit a multi-task elastic net (ℓ_1 ratio 0.9) with column-normalized regressors and per-unit standardized responses. Its regularization strength is chosen by five-fold cross-validation in which folds are formed by permutations, so that all rows generated under one permutation fall in the same fold, with the one-standard-error rule over ten log-spaced values in $[10^{-4}, 10^{-1}]$; if the chosen value leaves fewer than ten feature pairs with nonzero coefficients, we move toward weaker regularization until ten pairs (or all available pairs) have nonzero coefficients. Following a similar idea from stability selection (Meinshausen & Bühlmann, 2010), we repeat this estimation $K = 20$ times, each time on a random subset containing 75% of the permutations, so that pairs selected only under particular permutations receive low evidence. For each pair and layer/head, the proportion of fits in which its coefficient is nonzero is multiplied by the median absolute value of those nonzero coefficients (after undoing the column normalization); within each layer/head, these scores are ranked among feature pairs and the ranks are mapped to $[0, 1]$, with a layer/head whose scores are all zero left at zero. This is the evidence matrix E. A pair enters the rankings of Section 3.3 only if its two features fell into different groups in at least half of the permutations.

Algorithm A1 Feature-pair evidence from TabPFN attention (EVIDENCE)

Input: Dataset (X, y); number of permutations M; number of permutation subsets K
Output: Evidence matrix $E \in [0,1]^{P \times U}$ over the P feature pairs and U layer/head units; consensus scores c_p, $p = 1, \ldots, P$
$r \leftarrow$ out-of-fold residuals of an additive EBM fitted on (X, y)
for each feature permutation π_m, $m = 1, \ldots, M$ **do**
  $A_m \in \mathbb{R}^{U \times G \times G} \leftarrow$ feature-group attention of TabPFN fitted on $(X[:, \pi_m], r)$, where G is the number of feature groups $g_1, \ldots, g_G$ under π_m
  $Y_m \leftarrow$ upper triangle of each symmetrized, double-centered $G \times G$ matrix in A_m, one column per unit
  $D_m \leftarrow$ regressor rows of π_m: for group pair (a, b), entry $1/(|g_a||g_b|)$ at every feature pair (i, j) with $i \in g_a$, $j \in g_b$, then double-centered
end for
$D \leftarrow [D_1; \ldots; D_M] \in \mathbb{R}^{R \times P}$, $\quad Y \leftarrow [Y_1; \ldots; Y_M] \in \mathbb{R}^{R \times U}$, $\quad R = M\binom{G}{2}$
for $k = 1, \ldots, K$ **do**
  $\mathcal{S}_k \leftarrow$ a random subset of $\{1, \ldots, M\}$
  $B^{(k)} \leftarrow$ MULTITASKELASTICNET$(D_{\mathcal{S}_k}, Y_{\mathcal{S}_k}) \in \mathbb{R}^{P \times U}$
end for
$\phi_{p,u} \leftarrow \frac{1}{K} \sum_k \mathbf{1}\big[B^{(k)}_{p,u} \neq 0\big]$; $\quad \mu_{p,u} \leftarrow \mathrm{median}_k \big\{|B^{(k)}_{p,u}| : B^{(k)}_{p,u} \neq 0\big\}$
$E_{:,u} \leftarrow \mathrm{rankpct}(\phi_{:,u}\, \mu_{:,u})$ for each unit u; $\quad c_p \leftarrow \sqrt{\frac{1}{U} \sum_u E^2_{p,u}}$ for each pair p
return E, $\{c_p\}$

Algorithm A2 Pretraining the Global Learner

Input: Calibration datasets $\{(X_t, y_t)\}_{t=1}^{T}$
Output: Unit weights $\mathbf{w} \in \mathbb{R}^{U}_{\geq 0}$
for each calibration dataset $t = 1, \ldots, T$ **do**
  $E^{(t)}, \{c_p^{(t)}\} \leftarrow$ EVIDENCE(X_t, y_t)
  $\mathcal{C}_t \leftarrow$ the 30 pairs with the largest $0.6 \max_u E^{(t)}_{p,u} + 0.4\, c_p^{(t)}$
  for each candidate pair $p \in \mathcal{C}_t$ **do**
    $\mathrm{gain}_p \leftarrow$ held-out F1 gain of adding p to an additive EBM
  end for
  $\ell_p^{(t)} \leftarrow 1, 0.8, 0.6, 0.4, 0.2$ for the five best candidates with $\mathrm{gain}_p > 0$; $\quad \ell_p^{(t)} \leftarrow 0$ otherwise
end for
$\mathbf{E} \leftarrow \big[E^{(1)}_{\mathcal{C}_1,:}; \ldots; E^{(T)}_{\mathcal{C}_T,:}\big]$, $\quad \boldsymbol{\ell} \leftarrow \big[\ell^{(1)}_{\mathcal{C}_1}; \ldots; \ell^{(T)}_{\mathcal{C}_T}\big]$
$\mathbf{w} \leftarrow$ ELASTICNET$(\mathbf{E}, \boldsymbol{\ell})$ with nonnegative coefficients and no intercept
return $\mathbf{w}$

Calibration datasets. The Global Learner is trained on 30 binary classification datasets from the TabArena OpenML suite, which are disjoint from every PMLB dataset used for evaluation. Categorical columns are mapped to integer codes in order of first occurrence in the training set and missing or unseen values are imputed with the training median, with both mappings fitted on the training set. At most 40 features are kept, chosen by descending training variance. The out-of-fold residuals are computed on the full training set, and attention is measured on the 2,000 training rows with the largest $|r|$ (all rows if fewer). One dataset (OpenML 46939) is then excluded because no candidate pair improves the additive EBM on it, so 29 datasets contribute evidence and training labels.

Pretraining the Global Learner (Algorithm A2). For each calibration dataset t, we shortlist a candidate set $\mathcal{C}_t$ of the 30 pairs with the largest value of $0.6 \max_u E^{(t)}_{p,u} + 0.4\, c_p^{(t)}$, with $c_p^{(t)}$ as defined in Section 3.3. For each candidate we fit an EBM with that pair as its only interaction and an additive EBM on the same two stratified folds of the training set, using `max_bins`=256, `max_interaction_bins`=64, `outer_bags`=8, learning rate 0.02, 50 smoothing rounds, and at most 10,000 boosting rounds with early stopping after 75 rounds. The performance gain of a candidate is the difference in mean held-out F1 score. Candidates are ranked by gain; the top five with strictly positive gain receive training labels $\ell_p^{(t)} = 1, 0.8, 0.6, 0.4, 0.2$ in rank order, and all

Algorithm A3 Interaction Selection via TabDistill-Proxy

Input: Target dataset (X, y); number of interactions n; Global Learner $(\mathbf{w}, \gamma)$
Output: n pairwise interactions for a GAM model
$E, \{c_p\} \leftarrow \textsc{Evidence}(X, y)$
$e_p \leftarrow E_{p,:}\mathbf{w}$; $\quad b_p \leftarrow (1-\gamma)\,\mathrm{rankpct}(c_p) + \gamma\,\mathrm{rankpct}(e_p)$ for each pair p
$\mathcal{C} \leftarrow$ union of the ten highest-ranked pairs under b_p, c_p, and e_p
Reduce $\mathcal{C}$ to at most 16 pairs by three-fold drop-one validation of a joint EBM
Screen all subsets of $\mathcal{C}$ of size n by masking terms of the joint EBM; keep the 12 best
$S_0 \leftarrow$ the n highest-ranked pairs under b_p
$S_1, S_2 \leftarrow$ the two best screened subsets distinct from S_0
Refit S_0, S_1, S_2 with an independent three-fold evaluation
if some alternative S_i $(i \geq 1)$ has strictly higher pooled F1 than S_0 and wins on at least two folds **then**
  return the best such S_i
else
  return S_0
end if

remaining candidates receive a label of zero. Within each dataset, the pairs with nonzero labels and the pairs with zero labels are weighted so that each group has total weight 0.5. The evidence vectors and training labels of all 29 calibration datasets are then pooled, and an ElasticNet model with nonnegative coefficients and no intercept is fitted with ℓ_1 ratio 0.7 and regularization strength 0.3. The blending coefficient is fixed at $\gamma = 0.6$, so that $b_p = 0.4\,\mathrm{rankpct}(c_p) + 0.6\,\mathrm{rankpct}(e_p)$, where $\mathrm{rankpct}$ maps within-dataset ranks (ties averaged) to $[0, 1]$.

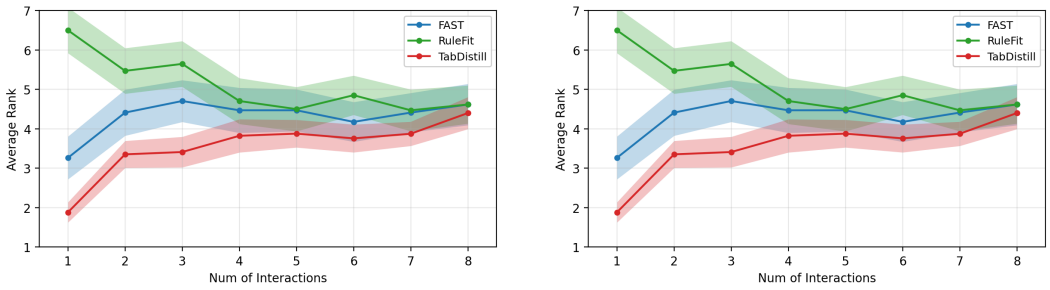
Selecting interactions on a target dataset (Algorithm A3). The target datasets are the binary classification datasets of our PMLB benchmark; categorical columns are ordinal-encoded. Residuals, attention measurements under $M = 60$ permutations, and the evidence matrix E are computed on the training set exactly as above. The union of the ten highest-ranked pairs under b_p, c_p, and e_p forms the candidate pool $\mathcal{C}$.

All EBMs in this stage use `outer_bags`=4 and `max_bins`=256, and the validation objective is F1 score. The selection relies on the additive structure of the EBM: for a model whose interactions are the candidate pool $\mathcal{C}$, the logit of each row decomposes as

$$\mathrm{logit}\,\hat{p}(x) = \underbrace{\beta + \sum_j f_j(x_j)}_{\text{main effects}} + \sum_{(i,j)\in\mathcal{C}} f_{ij}(x_i, x_j),$$

so the contribution of any subset of pairs can be removed from a fitted model by subtracting the corresponding f_{ij} terms, without refitting a separate EBM model.

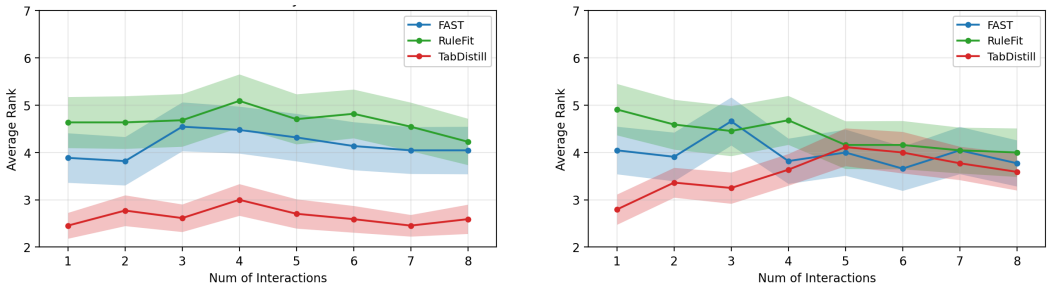
The three-fold EBM validation that reduces the candidate pool proceeds as follows: on three stratified folds we fit one EBM containing every pooled pair and score each pair by the drop in pooled out-of-fold F1 score when its term f_{ij} is removed from the held-out logits; the 20 best pairs are retained, the step is repeated, and the 16 best are kept, ordered by this score (pools of at most 16 pairs skip this step). To screen subsets of size $n \in \{1, \ldots, 8\}$, one EBM containing the 16 remaining candidate pairs is fitted on each training fold, and every $\binom{16}{n}$ subset S is scored by the pooled F1 score of the held-out logits with the terms of all pairs outside S removed (class 1 is predicted when the resulting logit is nonnegative); the 12 best subsets are kept; if two subsets tie, we choose the one whose pairs rank higher among the 16 remaining candidates. The final choice for budget n is made among at most three interaction sets: the default set, namely the n highest-ranked pairs under the blended score b_p, taken directly from the Global Learner's ranking, and up to two alternatives, namely the best distinct subsets found by screening the reduced pool. The independent three-fold evaluation refits each of these candidate sets on a separate stratified split. An alternative replaces the default set from the Global Learner only if its pooled F1 score is strictly higher and it wins on at least two of the three folds. The selected pairs are then passed to the downstream EBM and evaluated exactly as in the main experiments.



(a) Regression tasks evaluated by MSE.

(b) Regression tasks evaluated by R^2.

Figure A1: *Additional benchmark results in regression tasks.*



(a) Classification tasks evaluated by Accuracy.

(b) Classification tasks evaluated by AUROC.

Figure A2: *Additional benchmark results in classification tasks.*

Runtime measurement. Times in Figure 5(c) are wall-clock seconds at $n = 8$. For TabDistill-Proxy they cover residuals, attention measurements under 60 permutations, the evidence matrix, candidate selection, and the downstream EBM fit. For TabDistill they cover the SPEX search plus the downstream EBM fit. For FAST they are the EBM fit itself, since FAST selects interactions inside EBM training. All runs used a single NVIDIA GeForce RTX 5070 Ti (16 GB) for TabPFN and the CPU for EBM fits.

Table A1: *Summary table of TabDistill-Proxy.*

Component	Setting
TabPFN model	TabPFN-2 regressor, 1 estimator, input preprocessing off
Layer/head units	layers 6–11 $\times$ 6 heads ($U = 36$)
Out-of-fold residuals	additive EBM, 5 stratified folds, $r = y - \hat{p}$
Context/query split	two stratified halves, roles swapped; contrast weights on top 30% $\lvert r\rvert$ per class
Pair-contribution estimate	multi-task elastic net, ℓ_1 ratio 0.9, one-SE rule
Subsets of permutations	20 subsets of 75%
Eligible pairs	split across groups in $\geq 50\%$ of permutations
Calibration datasets	30 TabArena binary tasks (29 used); at most 40 features, by training variance
Rows measured per dataset	2,000 with largest $\lvert r\rvert$
Candidate pairs per dataset	30 by $0.6 \max_u E_{p,u} + 0.4\, c_p$
Performance gains	single-interaction vs. additive EBM, 2 folds, F1 score
ElasticNet	nonnegative, no intercept, ℓ_1 ratio 0.7, strength 0.3
Blending coefficient	$\gamma = 0.6$
Candidate pool	union of the top-10 pairs under b_p, c_p, and e_p
Subset screening	all $\binom{16}{n}$ subsets, keep 12

A.2 Additional Results for Main Experiments

We report the full rank breakdown for the main benchmark experiments. Across both regression and classification tasks, TabDistill in nearly all settings outperforms FAST and RuleFit over a wide range of interaction budgets.

Table A2: Regression tasks.

Metric	N_{int}	TabDistill	FAST	RuleFit
MAE	1	**1.82**	3.56	6.71
	2	**3.00**	4.68	5.94
	3	**3.26**	4.79	5.47
	4	**3.62**	4.18	5.21
	5	**3.27**	5.00	5.41
	6	**3.27**	4.68	5.15
	7	**4.03**	5.06	4.71
	8	**4.23**	4.91	4.71
MSE	1	**1.88**	3.26	6.50
	2	**3.35**	4.41	5.47
	3	**3.41**	4.71	5.65
	4	**3.82**	4.47	4.71
	5	**3.88**	4.47	4.50
	6	**3.76**	4.18	4.85
	7	**3.88**	4.41	4.47
	8	**4.40**	4.62	4.62
R^2	1	**1.88**	3.26	6.50
	2	**3.35**	4.41	5.47
	3	**3.41**	4.71	5.65
	4	**3.82**	4.47	4.71
	5	**3.88**	4.47	4.50
	6	**3.76**	4.18	4.85
	7	**3.88**	4.41	4.47
	8	**4.40**	4.62	4.62

Table A3: Classification tasks.

Metric	N_{int}	TabDistill	FAST	RuleFit
Accuracy	1	**2.45**	3.89	4.64
	2	**2.77**	3.82	4.64
	3	**2.61**	4.55	4.68
	4	**3.00**	4.48	5.09
	5	**2.70**	4.32	4.70
	6	**2.59**	4.14	4.82
	7	**2.45**	4.05	4.55
	8	**2.59**	4.05	4.23
F1 score	1	**2.77**	3.66	4.43
	2	**3.07**	4.14	4.70
	3	**2.91**	4.39	4.61
	4	**3.20**	3.89	4.73
	5	**3.05**	3.98	4.27
	6	**2.86**	3.68	4.18
	7	**2.82**	3.66	4.11
	8	**3.00**	3.66	3.89
AUROC	1	**2.80**	4.05	4.91
	2	**3.36**	3.91	4.59
	3	**3.25**	4.66	4.45
	4	**3.64**	3.82	4.68
	5	4.11	**4.00**	4.16
	6	4.00	**3.66**	4.16
	7	**3.77**	4.05	4.05
	8	**3.59**	3.77	4.00

Table A4: *Average rank of different models (lower is better).* Top: interpretable models. Bottom: black-box models. EBM, PyGAM, SNAM, and SIAN use the 4 interactions selected by TabDistill.

Model	MAE	MSE	R^2
EBM (with TabDistill)	3.44	3.74	3.74
PyGAM (with TabDistill)	7.77	7.85	7.85
SNAM (with TabDistill)	5.37	5.19	5.19
SIAN (with TabDistill)	3.63	3.74	3.74
NAM	8.22	8.19	8.19
Decision Tree	7.81	7.93	7.93
Linear Regression	7.11	6.74	6.74
NODE-GAM	3.67	3.44	3.44
XGBoost	5.78	5.85	5.85
TabPFN	2.11	2.26	2.26

A.3 Ablation Studies

We next report additional ablation results for different interaction indices and different SPEX query budgets (Figure A3).

A.4 TabPFN on Tree-Structured Interactions

We consider tree-structured teacher rules. We train a decision tree on synthetic classification data, replace the original labels with the tree's predictions, and then compare how well TabPFN and EBM recover the teacher's decision function. By increasing tree depth, we progressively increase the complexity of the induced interaction structure. As shown in Figure A4, TabPFN more faithfully

Table A5: Rank comparison across different numbers of interactions for different TabDistill index methods in regression tasks. Ranks are computed over the full set of models, including FAST and RuleFit. Lower is better.

Metric	N_{int}	TabDistill	FAST	RuleFit	BII	Fourier	FSII	Mobius	SII	STII
MAE	1	1.82	3.56	6.71	1.91	1.61	1.76	**1.56**	1.79	1.62
	2	3.00	4.68	5.94	2.79	3.09	3.00	2.91	**2.74**	3.00
	3	3.26	4.79	5.47	4.29	4.28	**3.12**	3.18	3.76	3.47
	4	3.62	4.18	5.21	5.24	4.97	3.62	3.65	4.91	**3.44**
	5	**3.27**	5.00	5.41	5.06	5.00	3.76	3.71	4.88	3.88
	6	**3.27**	4.68	5.15	6.10	5.07	3.55	4.27	5.44	3.48
	7	**4.03**	5.06	4.71	5.19	4.46	4.52	4.33	4.61	4.33
	8	4.23	4.91	4.71	4.60	5.68	4.19	4.48	4.20	**4.06**
MSE	1	1.88	3.26	6.50	1.74	1.55	1.82	2.00	**1.47**	2.12
	2	3.35	4.41	5.47	2.71	**2.55**	3.35	3.18	2.68	3.44
	3	3.41	4.71	5.65	4.29	4.16	**3.09**	3.15	3.67	3.50
	4	3.82	4.47	4.71	5.03	4.68	**3.65**	4.06	4.48	3.88
	5	**3.88**	4.47	4.50	4.70	5.16	4.09	4.15	4.59	4.44
	6	3.76	4.18	4.85	5.77	5.50	4.06	4.42	5.03	**3.48**
	7	**3.88**	4.41	4.47	5.23	5.12	4.85	4.76	4.61	4.09
	8	4.40	4.62	4.62	4.73	5.91	4.35	4.87	4.27	**3.42**
R^2	1	1.88	3.26	6.50	1.74	1.55	1.82	2.00	**1.47**	2.12
	2	3.35	4.41	5.47	2.71	**2.55**	3.35	3.18	2.68	3.44
	3	3.41	4.71	5.65	4.29	4.16	**3.09**	3.15	3.67	3.50
	4	3.82	4.47	4.71	5.03	4.68	**3.65**	4.06	4.48	3.88
	5	**3.88**	4.47	4.50	4.70	5.16	4.09	4.15	4.59	4.44
	6	3.76	4.18	4.85	5.77	5.50	4.06	4.42	5.03	**3.48**
	7	**3.88**	4.41	4.47	5.23	5.12	4.85	4.76	4.61	4.09
	8	4.40	4.62	4.62	4.73	5.91	4.35	4.87	4.27	**3.42**

Table A6: Rank comparison across different numbers of interactions for different TabDistill index methods in classification tasks. Ranks are computed over the full set of models, including FAST and RuleFit. Lower is better.

Metric	N_{int}	TabDistill	FAST	RuleFit	BII	Fourier	FSII	Mobius	SII	STII
Accuracy	1	2.45	3.89	4.64	**2.02**	2.52	2.64	2.66	2.34	2.27
	2	2.77	3.82	4.64	2.77	2.77	2.50	2.50	**2.11**	2.55
	3	**2.61**	4.55	4.68	3.09	2.88	2.75	2.64	**2.61**	2.75
	4	3.00	4.48	5.09	2.52	2.91	3.18	**2.39**	2.91	2.41
	5	2.70	4.32	4.70	2.98	3.60	2.68	2.61	2.75	**2.45**
	6	2.59	4.14	4.82	2.95	3.17	2.80	2.57	3.02	**2.50**
	7	**2.45**	4.05	4.55	2.77	3.17	2.70	3.18	2.89	2.95
	8	**2.59**	4.05	4.23	2.98	3.34	2.84	2.89	2.93	2.73
F1 score	1	2.77	3.66	4.43	**2.23**	2.89	2.95	3.00	2.75	2.57
	2	3.07	4.14	4.70	2.68	3.27	2.89	2.75	**2.45**	2.86
	3	2.91	4.39	4.61	3.25	3.40	2.93	**2.77**	2.98	2.98
	4	3.20	3.89	4.73	**2.70**	3.33	3.50	2.75	3.32	2.86
	5	3.05	3.98	4.27	3.02	3.81	3.14	3.30	3.07	**2.91**
	6	**2.86**	3.68	4.18	3.45	3.49	3.27	3.00	3.52	2.89
	7	**2.82**	3.66	4.11	3.26	3.55	3.14	3.36	3.41	3.20
	8	**3.00**	3.66	3.89	3.37	3.21	3.23	3.27	3.42	3.11
AUROC	1	2.80	4.05	4.91	2.52	3.18	**2.41**	2.80	2.80	2.59
	2	3.36	3.91	4.59	2.95	3.73	2.98	3.18	**2.80**	3.27
	3	3.25	4.66	4.45	3.30	4.07	3.30	3.18	3.41	**2.95**
	4	3.64	3.82	4.68	3.45	4.47	3.41	**3.11**	3.48	3.32
	5	4.11	4.00	4.16	3.91	4.71	3.50	3.50	3.52	**3.11**
	6	4.00	3.66	4.16	3.95	4.44	3.91	**3.39**	3.52	3.59
	7	3.77	4.05	4.05	3.88	4.33	4.02	3.68	**3.36**	3.84
	8	**3.59**	3.77	4.00	3.88	4.34	3.77	3.77	3.93	3.86

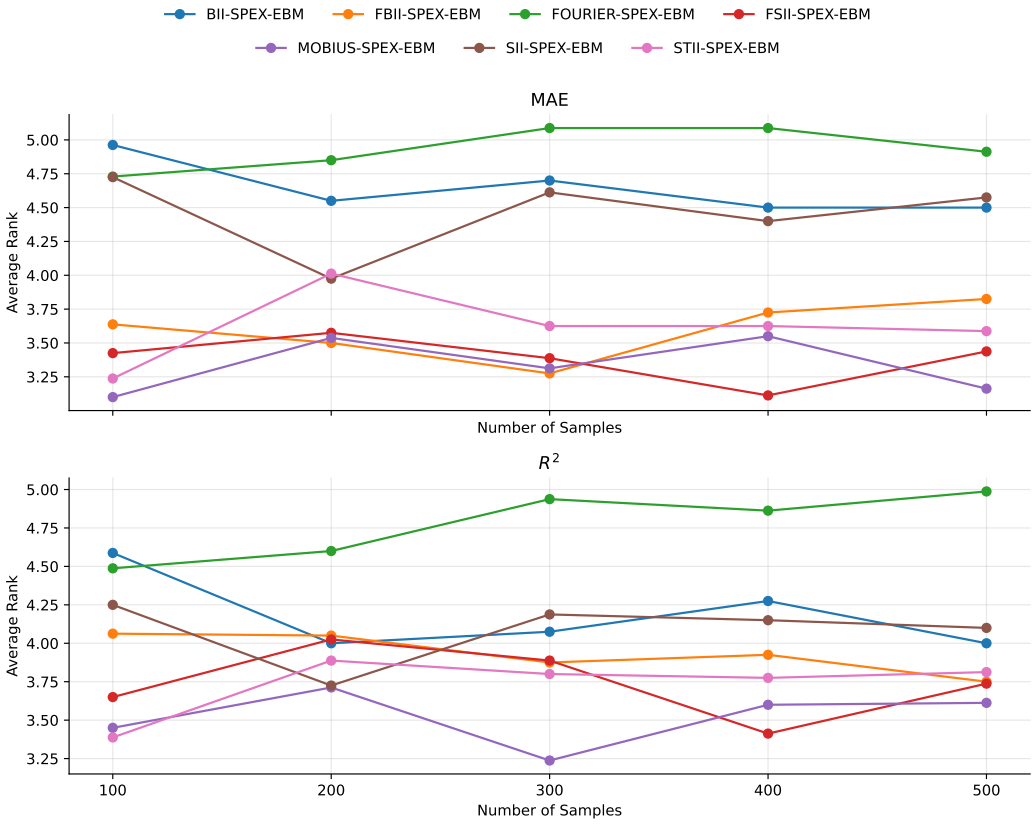


Figure A3: *Stability of interaction selection with respect to the SPEX query sample size.* We vary the query sample size used by SPEX from 100 to 500 and evaluate downstream predictive performance on TabArena regression datasets, reported as average rank across datasets. Across all indices, rank variations remain within approximately 0.5.

matches the teacher across depth levels, indicating that its interaction-learning advantage extends beyond Fourier-sparse functions to non-smooth tree-based rules.

Dataset Generation: We generated synthetic binary classification data using `make_classification` with $n = 10000$ samples and $p = 15$ features, of which $p_{\text{inf}} = 10$ were informative. The data were randomly split into training and test sets with a 7:3 ratio. To construct tree-structured target functions, we first trained a decision tree classifier on the training set using the original labels, and then replaced the original labels with the tree's predictions. This yields pseudo-labels $\tilde{y} = f_{\text{tree}}(X)$, where f_{tree} denotes the fitted decision tree. The student models are evaluated on how well they recover the teacher's decision rule, rather than the original data-generating labels.

Experiment – Tree-Structured Rules: We varied the maximum depth of the teacher tree over $D \in \{1, 2, \ldots, 10\}$. Increasing D produces progressively more complex and non-smooth decision boundaries, corresponding to higher-order and more localized interaction structures. For each depth, we trained student models on the training set with pseudo-labels $(X_{\text{train}}, \tilde{y}_{\text{train}})$ and evaluated them on the testing set with pseudo-labels $(X_{\text{test}}, \tilde{y}_{\text{test}})$.

Baselines: We compared TabPFN against EBM as a student model for recovering the teacher rule. EBM was trained with budget set `num_interactions` to p_{inf}, using `outer_bags`$= 4$ and `max_bins`$= 256$. We ran the FAST algorithm for interaction detection in EBM. Performance was measured using classification accuracy, AUROC, and F1 score.

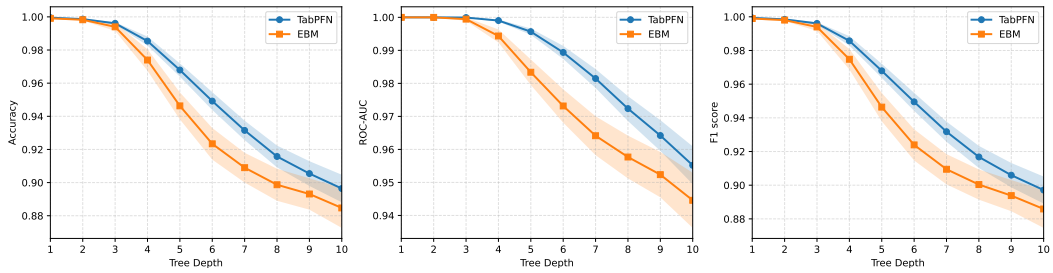


Figure A4: We simulate tree-structured data with $n = 10{,}000$ samples and $p = 15$ features, varying the decision tree depth from 1 to 10. As the decision boundary becomes increasingly complex and non-smooth, TabPFN more faithfully approximates data generated from tree-structured decision rules. All results are averaged over 20 random seeds.

A.5 Interaction Selection in the Fiat 500 Car Price Prediction Case Study

Setup. The dataset (*Another Dataset on used Fiat 500 (1538 rows)*, OpenML task ID 363615) contains 1,538 samples, each corresponding to an individual vehicle listing collected from a specialized second-hand car marketplace. It includes seven predictor variables, comprising both continuous and discrete features that describe vehicle characteristics and seller-related information; the prediction target is the vehicle's selling price. We use the official training/testing split provided by the OpenML task (train–test ratio 2:1) and evaluate on the test set. For each method, we select the top 10 interactions and train an EBM with the selected interaction set, using `outer_bags`=4 and `max_bins`=256 and default values for all other hyperparameters.

Table A7: Performance comparison for the Fiat 500 price prediction task.

Method	MSE	RMSE	MAE	R^2
TabDistill	509888	714.1	536.1	0.8595
FAST	540500	735.2	540.0	0.8511
RuleFit	523756	723.7	545.9	0.8557

Results. Table A7 reports predictive performance and Table A9 lists the selected interactions. TabDistill captures higher-order interactions between geographic location and vehicle characteristics, including vehicle model and age. In contrast, FAST is limited to pairwise effects and therefore captures only partial location-related interactions. Although RuleFit can identify higher-order interactions, it fails to capture latitude and longitude jointly, so it misses the seller's geographic location in this task. Figure 7 and Table A8 visualize the `(longitude, latitude, model_type)` interaction discussed in the main text.

Table A8: *Median selling price (in $k) by model type across* 2×2 *location cells.* The four location cells are defined by splitting longitude and latitude at their median values.

	Model type	
Location cell	Sport	Non-sport
South-West	5.90	9.40
South-East	5.80	9.30
North-West	5.70	9.39
North-East	5.50	9.80

Table A9: Comparison of selected feature interactions on the used car pricing task.

	TabDistill	**FAST**	**RuleFit**
1	(vehicle model, vehicle age)	(vehicle model, vehicle age)	(vehicle age, mileage)
2	(vehicle model, vehicle age, mileage)	(vehicle model, latitude)	(vehicle age, mileage, longitude)
3	(vehicle age, mileage)	(vehicle model, longitude)	(mileage, longitude)
4	(vehicle age, latitude)	(engine power, latitude)	(vehicle age, latitude)
5	(vehicle model, vehicle age, latitude, longitude)	(vehicle age, mileage)	(vehicle age, mileage, latitude)
6	(vehicle model, vehicle age, latitude)	(vehicle age, latitude)	(vehicle model, vehicle age, mileage)
7	(vehicle model, mileage)	(vehicle age, longitude)	(vehicle model, vehicle age, mileage)
8	(vehicle model, vehicle age, longitude)	(mileage, latitude)	(vehicle model, vehicle age, mileage)
9	(vehicle model, latitude)	(mileage, longitude)	(vehicle model, vehicle age, mileage)
10	(vehicle model, latitude, longitude)	(latitude, longitude)	(vehicle model, vehicle age, mileage)