Finding Structure Without Fitting Noise: Taking Apart the Small-Table Advantage of Tabular Foundation Models

Anonymous authors
Paper under double-blind review

Abstract

Tabular foundation models (TFMs) beat tuned gradient boosting and BART on small tables, but an accuracy gap cannot tell finding more structure from fitting less noise. We take a TFM's advantage over a comparator apart with three black-box instruments that need only fit-and-predict calls. *Conditional twins* keep a table's least-squares fit and residual size and redraw the residual at random, which removes the structure in the residual but keeps everything the linear fit reveals. A *structure ablation* removes structure estimated on rows outside the context and measures how much of it each learner had found. *Noise redraws* resample only the noise around that structure and measure what fitting it costs. On 57 OpenML regression tables with 120 context rows, all 11 TFMs we audit have a better mean rank than each of 23 comparators. The four we study in depth find more structure than the tuned boosters on most tables, and more than BART, with the largest gaps on the tables with the most structure: removing the structure costs them more accuracy than BART on 37 of 52 tables, and summed over tables they recover 42% of what it adds to a linear fit against BART's 23%, including structure beyond an additive fit that BART barely uses. They also fit less noise than CV-tuned boosting and LightGBM: the booster's extra cost of fitting noise is 15–20% of their advantage over CV-tuned boosting, and averaging the booster's fold models removes most of it while the TFMs keep 86% of their advantage.

1 Introduction

Tabular foundation models (TFMs) predict a new row by reading a labeled table as context, with parameters fixed by pretraining on synthetic tasks or real tables (Hollmann et al., 2025; Qu et al., 2025; 2026; Hosseinzadeh et al., 2026). On tables with a few hundred rows they beat gradient boosting tuned by cross-validation (Hollmann et al., 2025; Grinsztajn et al., 2026). The usual explanation is that a TFM approximates Bayesian posterior prediction under its pretraining prior (Müller et al., 2022; Hollmann et al., 2025), an account supported on synthetic data, where the prior is known (Müller et al., 2022). On a real table, however, an accuracy gap is a single number. It cannot say whether the TFM found more of the table's structure or fitted less of its noise, and the two call for different responses: a comparator's excess noise can be averaged away, while structure it misses cannot.

Approach. We take the advantage apart with three black-box instruments, each answering one question (Figure 3).

- *Conditional twins: what does a learner make of the linear fit alone?* A twin is a copy of the table with the same features, the same least-squares fit and the same residual size, and a residual drawn at random (Figure 2). Averaged over twins, a learner's forecast depends on the targets only through what the linear fit reveals: its coefficients and the residual's size.
- *Structure ablation: how much of the structure beyond the linear fit does a learner find?* We estimate that structure on spare rows (rows not used as context, test or audit rows), remove it from the context, and measure how much accuracy each learner loses.
- *Noise redraws: what does fitting noise cost a learner?* Holding the estimated structure fixed, we redraw only the noise around it and measure how much the learner's forecasts move.

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.

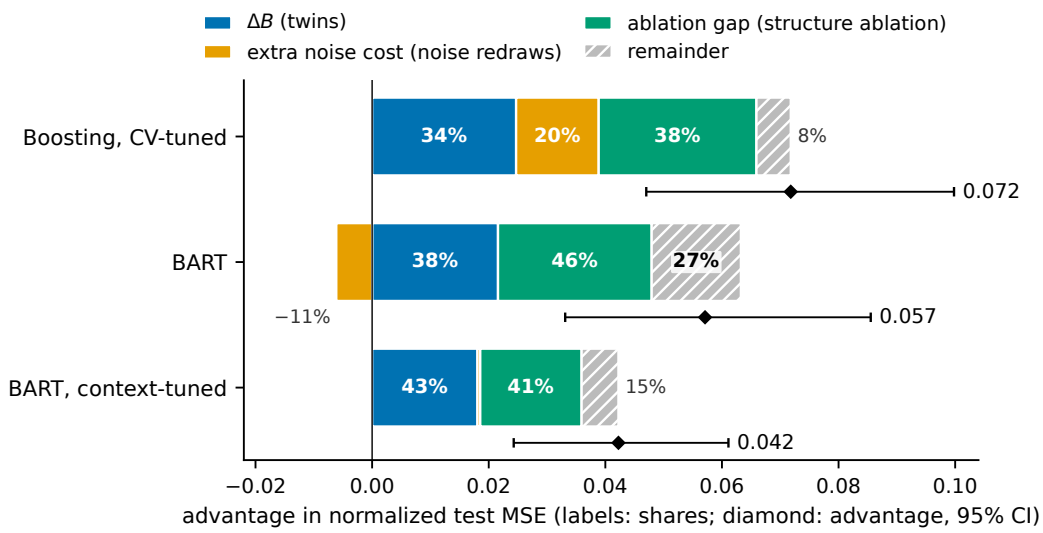

Figure 1: The TFMs' advantage over three comparators, split as in Eq. (6). Against CV-tuned boosting the TFMs both find more structure (green) and fit less noise (orange); against BART, which fits less noise than the TFMs (orange, left of zero), nearly half of the advantage is extra discovery (the ablation gap). The linear-fit part ΔB (blue) compares forecasts averaged over twins, which keep the features, the linear fit and the residual's size; the remainder (hatched) is defined in Section 2.4. Labels give each part's share of the advantage; the diamond is the advantage with its 95% interval. Spare-row tables with noise redraws (Tables 17 and 20).

All three need only fit-and-predict calls, so they apply to any learner. Together they split the advantage into parts (Figure 1).

Findings. We audit 11 TFMs and 23 comparators on 57 OpenML regression tables with 120 context rows; all eleven TFMs have a better mean rank than every comparator. The *four main TFMs*, which we run as single estimators, are TabPFN 2.6, TabPFN 3, TabICLv2 and TabDPT-Turbo; "the TFMs" means these four, with their losses averaged.

1. **The TFMs find more structure, most where there is most to find.** When structure estimated on spare rows is removed from the context, the TFMs lose a larger share of the accuracy it adds to a linear fit than BART, context-tuned BART and three tuned boosters (Table 2). Against the tuned boosters the gap is broad, on most tables; against BART it is concentrated on the spare-row tables with the most structure, and it replicates on 14 of 16 OpenML-CTR23 tables that no model lists in its pretraining data. The TFMs use structure beyond an additive fit, which BART barely uses. On planted structure their edge over BART is larger for interactions than for smooth structure, and negative for steps with heteroscedastic noise. On structure drawn from BART's prior family, BART recovers more, so the ablation does not simply favor the TFMs (Section 4.1).
2. **They fit less noise than CV-tuned boosting and LightGBM.** The booster's extra cost of fitting noise is 15–20% of the TFMs' advantage over CV-tuned boosting (orange), and similar against Optuna-tuned LightGBM. Averaging CV-tuned boosting's fold models removes most of it, and the TFMs keep 86% of their advantage (Section 4.2).

Most of the rest is the linear-fit part ΔB (blue), about a third of the advantage over CV-tuned boosting; over boosting and BART it persists when the truth is exactly linear (Section 4.3).

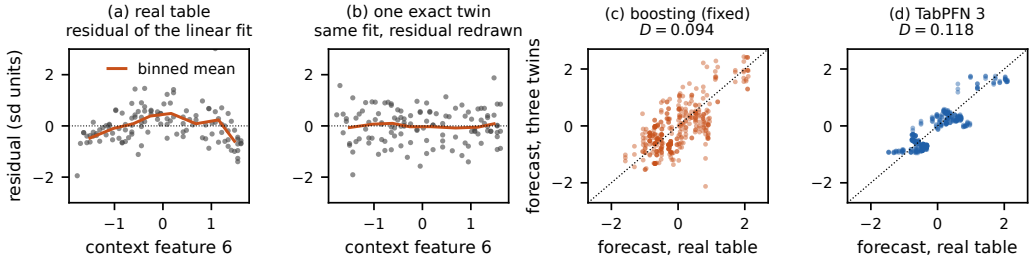

Figure 2: Twins on one real table (OpenML *analcatdata_vineyard*, 120 context rows). **(a)** The residual of the linear fit carries structure (binned mean in orange). **(b)** A twin keeps the same fit and residual size but redraws the residual at random, so it carries no structure. **(c)** Gradient boosting (fixed settings): its test forecasts from three twins scatter widely against its real-table forecasts. **(d)** TabPFN 3's forecasts vary little from twin to twin. Net discovery D is how much better a learner forecasts from the real table than from its twin average $\bar{p}$; it is lower for boosting, whose forecasts carry more fitted noise, which averaging over twins removes from $\bar{p}$ but not from the real-table forecast.

2 Method

2.1 Conditional twins

A twin is the table as a linear-Gaussian reference model would regenerate it: the same features, the same linear fit and the same residual size, but a residual drawn at random, so that it carries no structure. Formally, let $X \in \mathbb{R}^{n\times d}$ be the context features and $Z = [1, X]$ the design matrix. The reference model is

$$y = Z\theta + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I_n), \tag{1}$$

for which $T(y) = (Z^\top y,\ y^\top y)$ is sufficient. With P_Z the projector onto $\mathrm{col}(Z)$ and $r = (I - P_Z)y$, a twin is

$$y' = P_Z y + \|r\|\, u, \qquad u \text{ uniform on the unit sphere of } \ker(Z^\top). \tag{2}$$

Proposition 1 (Exact conditional twin; Langsrud, 2005; Lindqvist & Taraldsen, 2005) *Suppose* $\mathrm{rank}(Z) \le n-2$. *Then* $T(y') = T(y)$, *and under Eq.* (1), y' *is a draw from the conditional law of* y *given* X *and* $T(y)$, *conditionally independent of* y *given* $T(y)$.

We use $K = 39$ twins per table.

2.2 What the twins measure

Each table is split into a *partition* of 120 context rows, 120 test rows and 64 audit rows (used to screen partitions in Section 2.3). Let $p_k(x)$ be a learner's forecast at test row x with twin k as context ($k = 0$ the real table) and $\bar{p}(x) = K^{-1}\sum_{k\ge 1} p_k(x)$ its twin average. The test loss $L_t(p)$ is the mean squared error divided by the context variance v of the target (normalized MSE), and L_t^{ls} is the loss of the linear fit. We report

$$\text{net discovery:} \quad D = L_t(\bar{p}) - L_t(p_0), \tag{3}$$

$$\text{linear-fit term:} \quad B = L_t(\bar{p}) - L_t^{\mathrm{ls}}. \tag{4}$$

The loss relative to the linear fit then splits exactly as

$$L_t(p_0) - L_t^{\mathrm{ls}} = B - D. \tag{5}$$

The twin average $\bar{p}$ depends on the targets only through $T(y)$: averaging over twins removes whatever the learner reads from the direction of the residual, which is random on a twin. B therefore compares least squares with the learner's forecast from the features and what the linear fit reveals; $B < 0$ means that forecast beats least squares. D compares the real-table forecast with $\bar{p}$. Structure in the real

Algorithm 1 The twin audit of one learner on one table

Require: context (X, y) with reference basis Z, test rows (X_t, y_t) with basis Z_t, learner f, K twins

$\hat{\theta} \leftarrow (Z^\top Z)^+ Z^\top y,\ \hat{y} \leftarrow Z\hat{\theta},\ r \leftarrow y - \hat{y}$ ▷ linear fit and residual on the context

for $k = 1, \ldots, K$ **do**

  draw u_k uniformly on the unit sphere of $\ker(Z^\top)$; $y^{(k)} \leftarrow \hat{y} + \|r\|\, u_k$ ▷ twin k

  $p_k \leftarrow f(X, y^{(k)})(X_t)$

$p_0 \leftarrow f(X, y)(X_t),\ \bar{p} \leftarrow K^{-1} \sum_k p_k$

$L_t^{\text{ls}} \leftarrow L_t(Z_t\hat{\theta}),\ D \leftarrow L_t(\bar{p}) - L_t(p_0),\ B \leftarrow L_t(\bar{p}) - L_t^{\text{ls}}$

return D, B, with $L_t(p_0) - L_t^{\text{ls}} = B - D$

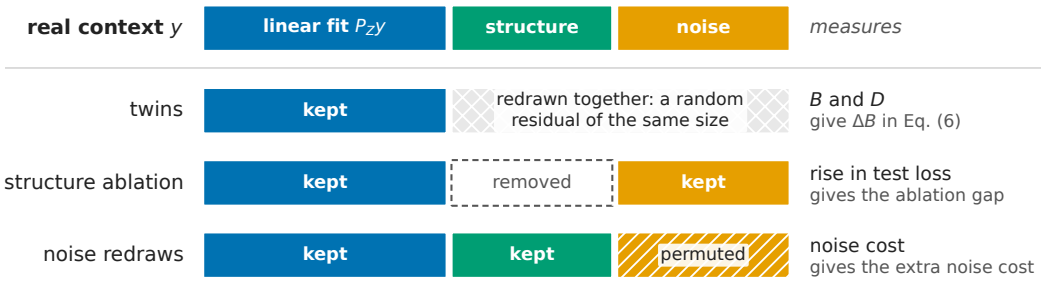

Figure 3: What each instrument keeps and changes in the context targets $y = P_Z y + (I - P_Z)\hat{f}(X) + (I - P_Z)e$ (the linear fit, the structure beyond it and the noise, all that $\hat{f}$ misses) and what it measures. Twins run on all tables, the ablation and the noise redraws on the spare-row tables (Section 2.3).

residual is missing from the twins, so D rises with the structure a learner uses; noise a learner fits moves p_0 but averages out of $\bar{p}$, so D falls with it. Two exact results support this reading. Without structure (under Eq. (1)) the real table is exchangeable with its twins, so for any learner the expected D is minus the expected variance of its forecasts across twins, a pure noise cost (Appendix A). For a linear smoother (forecasts linear in the context targets) under homoscedastic noise, the expected D is the learner's *discovery*, the reduction in test loss from the structure it uses, minus its *noise cost*, the loss it incurs by fitting noise (Proposition 2, in unnormalized MSE, as $K \to \infty$). The TFMs, BART and boosting are not linear smoothers, so we measure both parts directly (Section 2.3).

Comparing learners. The TFMs' *advantage* over a comparator is the comparator's real-table test loss $L_t(p_0)$ minus theirs; by Eq. (5) it splits exactly as $\Delta B + \Delta D$. Every gap Δ between the TFMs and a comparator is signed so that positive favors the TFMs (ΔB is the comparator's B minus the TFMs', ΔD the TFMs' D minus the comparator's). Algorithm 1 needs $K + 1$ fit-and-predict calls per learner and table; Table 1 lists the quantities.

2.3 Structure ablation and noise redraws

These use the 36 re-partitioned tables with at least 1,000 rows outside the partition (*spare rows*; the *spare-row tables*). On the spare rows we fit $\hat{f}$, the average of a random forest and gradient boosting.

Discovery. We remove $\hat{f}$'s part beyond the linear fit from the context targets, $y^- = y - (I - P_Z)\hat{f}(X)$, which leaves the linear fit unchanged. The test rows still contain the structure, so a learner that had used it loses accuracy when it is removed from the context, and a learner whose forecasts ignored it loses nothing. A learner's rise in test loss from y to y^-, $L_t(p_0^-) - L_t(p_0)$ with p_0^- its forecast from y^-, thus estimates its discovery (for an adaptive learner, plus any change in the noise it fits; Appendix I separates the two); the TFMs' rise minus a comparator's is the *ablation gap*. A learner's *share recovered* is its rise divided by the structure's *worth*, $L_t(Z_t\hat{\theta}) - L_t(Z_t\hat{\theta} + h_t)$: how much adding $\hat{f}$'s part beyond the linear fit at the test rows, $h_t = \hat{f}(X_t) - Z_t\theta_{\hat{f}}$ with $\theta_{\hat{f}}$ the least-squares

Table 1: The quantities in the body, all in normalized test MSE. A gap Δ between the TFMs and a comparator is signed so that positive favors the TFMs.

Quantity	Instrument	Meaning
B, linear-fit term	twins	loss of the twin-averaged forecast minus that of least squares
D, net discovery	twins	loss of the twin-averaged forecast minus that of the real-table forecast
noise cost	noise redraws	forecast variance across redraws of the noise around $\hat{f}$
rise	ablation	increase in test loss when $\hat{f}$'s structure is removed from the context
worth	ablation	decrease in the linear fit's test loss when $\hat{f}$'s structure is added to it
share recovered	ablation	rise divided by worth
ablation gap	ablation	the TFMs' rise minus the comparator's
remainder	Eq. (6)	ΔD minus the extra noise cost and the ablation gap

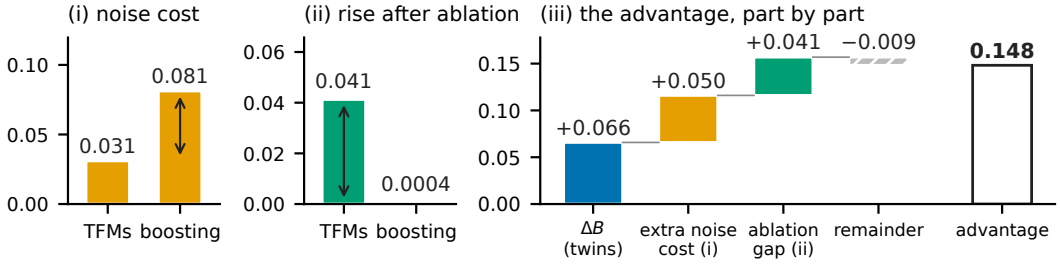

Figure 4: Worked example: the TFMs' advantage over CV-tuned boosting on *physicochemical-protein* (9 partitions). (i) Noise cost; the arrow is the booster's extra noise cost. (ii) Rise in test loss when the structure is removed; the arrow is the ablation gap. (iii) Eq. (6): ΔB, the two arrows and the remainder add up to the advantage.

coefficients of $\hat{f}(X)$ on Z, lowers the linear fit's test loss. Partitions where $\hat{f}$ does not improve the linear fit on the audit rows are skipped, since there is then no evidence of structure.

Noise cost. With $\hat{f}$ fixed, we redraw only the rest of the context targets, $e = y - \hat{f}(X)$, by permuting it: $y_j = P_Z y + (I - P_Z)\hat{f}(X) + (I - P_Z)e_{\pi_j}$ for nine permutations π_j of the rows. We measure a learner's noise cost as the variance of its test forecasts across the y_j (averaged over test rows and divided by v), since a learner's forecast variance across draws of the noise adds one for one to its expected test loss (Appendix I).

2.4 Splitting the advantage

A learner's rise is its loss on y^- minus its loss on y, so the ablation gap is exactly the TFMs' advantage on y minus their advantage on y^-, the advantage *left* once the structure is removed:

$$\text{advantage} = \text{ablation gap} + \text{advantage left}.$$

With the structure gone, what is left should be the linear-fit part ΔB plus the comparator's *extra noise cost* (its noise cost minus the TFMs'). When both learners are linear smoothers under homoscedastic noise and $\hat{f}$ is the true regression function, this holds in expectation (unnormalized, as $K \to \infty$; Appendix A): each twin average depends only on the linear fit, which removing the structure keeps, and with no structure left the only remaining difference in net discovery is how much noise each learner fits. We therefore write

$$\text{advantage} = \underbrace{\Delta B}_{\text{twins}} + \underbrace{\text{extra noise cost}}_{\text{noise redraws}} + \underbrace{\text{ablation gap}}_{\text{ablation}} + \text{remainder}. \tag{6}$$

Eq. (6) defines the *remainder*. Since the advantage is exactly $\Delta B + \Delta D$, the remainder is ΔD minus the extra noise cost and the ablation gap: the part of the twins' net-discovery gap that the two direct measurements do not account for. The split into ΔB and ΔD is thus exact, and the extra noise cost

and the ablation gap are measurements, each checked against a planted truth (Appendices B and I). Figure 4 follows the split on one table, where the four main TFMs beat CV-tuned boosting by 0.148: boosting fits more noise (panel i) and had used almost none of the structure the TFMs used (panel ii). Over the spare-row tables the remainder is 8% of the advantage over CV-tuned boosting and 27% over BART (Figure 1). Against BART it means that, after the estimated structure is removed, the TFMs still gain more than BART from what the residual carries. Relative to noise redraws that flip only the signs of the residual around $\hat{f}$, most of the TFMs' gain from the real residual persists (Appendix I), consistent with structure $\hat{f}$ missed rather than with how the noise's size varies across rows.

3 Setup

Tables. We use 57 OpenML regression tables (Table 15; Appendices E and F). Of these, 34 were drawn at random from OpenML's 3,081 regression tasks after excluding every dataset that a model lists in its pretraining data; the others are widely used OpenML regression tables. The drawn tables lie outside the pretraining data of ten of the eleven TFMs; the eleventh, TabDPT-Turbo, describes its corpus only by a rule over OpenML. To replicate the ablation we add the 16 tables of the OpenML-CTR23 benchmark (Fischer et al., 2023) that are not in our roster and have at least 1,000 spare rows (the *new CTR23 tables*); TabDPT-Turbo's corpus excludes the CTR23 datasets (Hosseinzadeh et al., 2026), so none of the models lists them in its pretraining data. Preprocessing (median imputation, one-hot coding, standardization, at most 20 columns, so that the linear fit leaves most of the 120 rows' degrees of freedom to the residual) is computed from the context features, and every learner receives the same numeric matrix (Appendix E).

Learners. Eleven TFMs, one released checkpoint each: TabPFN 2.6 (Grinsztajn et al., 2025; Prior Labs, 2026), TabPFN 3 (Grinsztajn et al., 2026), TabICLv2 (Qu et al., 2026), TabDPT-Turbo 1.2 (Hosseinzadeh et al., 2026), TabFM 1.0 (Google Research, 2026), TabSwift (Liu & Ye, 2026), MITRA v1 (Zhang et al., 2025), LimiX-2M (Wang et al., 2026), Xiaomi-TabLDM (Xiaomi-TabLDM Team, 2026), EXAONE-Tabular v1 (Eo et al., 2026) and Nori-30M (Synthefy et al., 2026). The first four run as single estimators and the other seven with eight ensemble members where available (Appendix E). The 23 comparators (Appendices I and J) include BART (Chipman et al., 2010) at its defaults and context-tuned, DART (Linero, 2018), five Gaussian processes (one with marginalized hyperparameters; Rasmussen & Williams, 2006), CV-tuned scikit-learn histogram gradient boosting (Pedregosa et al., 2011), LightGBM tuned by 50 Optuna trials (Ke et al., 2017; Akiba et al., 2019), CatBoost (Prokhorenkova et al., 2018) at its defaults and early-stopped on a validation split of the context rows (and, in the ablation, CV-tuned), a random forest, MLPs, k-NN, CV ridge (Pedregosa et al., 2011), RealMLP, TabM (Holzmüller et al., 2024; Gorishniy et al., 2025) and a GAM (Wood, 2011). *CV-tuned* and *context-tuned* mean hyperparameters chosen on the context rows alone.

Evaluation. All learners share the test rows of one partition per table (the *main partition*), which gives the ranks. The TFMs, BART, context-tuned BART, DART, the marginalized GP and CV-tuned boosting also run on up to nine more partitions of each table (Appendix K), and comparisons among them average each table's partitions. The body thus uses three sets of tables: all 57 for the ranks, the 36 spare-row tables for the ablation (35 of them for the split and the noise redraws), and the new CTR23 tables to replicate the ablation (Table 20 gives the partitions behind each result), so values such as the advantage over BART differ somewhat between them. We report means over tables with 95% bootstrap intervals (5,000 resamples), shares as ratios of these means, and the number of tables on which the TFMs come out ahead, with a sign test. One audit of a main TFM on one table takes a median of 2–27 s on one GPU (Appendix N).

4 Results

On the main partition all eleven TFMs have a better mean rank than every comparator (Appendix C). Figure 1 splits the four main TFMs' advantage over CV-tuned boosting, BART and context-tuned BART; we take the parts in turn.

4.1 The TFMs find more structure

In total. Over all 52 spare-row and new CTR23 tables, removing the structure raises the TFMs' loss more than BART's on 37 ($p = 0.003$), and summed over tables the TFMs recover 42% of the structure's worth and BART 23% (Table 2). The mean gap's interval excludes zero on each set alone, with the seven other TFMs in place of the four, with TabPFN 3 instead of trees estimating the structure, and against context-tuned BART and each tuned booster. Against BART the gap is nearly half of the TFMs' advantage (46% [20, 64]; Figure 1).

Table by table. Against the tuned boosters the gap is broad: the TFMs lose more than each of them on 28 of the 36 spare-row tables ($p = 0.001$). Against BART it is concentrated where there is most to find. On the spare-row tables the TFMs lose more than BART on 23 of 36 ($p = 0.132$), and the five largest gaps, all among the seven tables whose structure is worth most, carry 84% of the summed gap; without them the mean gap is 0.006 (Figure 5a). The largest is on *CERN-Electron-Collision-Data*, whose target, an invariant mass, is a smooth function of products of the recorded energies and momenta. On the five spare-row tables where the TFMs' advantage is largest, removing the structure takes it from 0.202 to 0.007 (0.182 to 0.035 when the five are chosen on half of each table's partitions and scored on the rest). On the new CTR23 tables the gap is larger and more consistent, positive on 14 of 16 ($p = 0.004$; Figure 5b).

Controls. The ablation recovers the true gap on planted structure, is not an effect of the smaller spread of the ablated targets (BART with its prior scales frozen at their real-context values changes little), and does not simply favor the TFMs: BART and the marginalized GP each recover more than the TFMs on structure drawn from their own prior family (Figure 5d; all checks in Table 5).

What the TFMs find. We split the estimated structure into its fitted additive approximation (a sum of one-feature functions, fitted to $\hat{f}$ by boosting without feature interactions) and the rest, which holds what interactions among features add and any additive structure the approximation misses, and remove each part on its own. The TFMs' gap over BART is uncertain for the additive part but positive for the rest, which BART barely uses (Figure 5c; Table 6). With structure of a known kind planted on the real feature matrices of twelve roster tables, the TFMs' true gap over BART is 0.068 for interaction-heavy structure, 0.015 for smooth structure and -0.016 for step functions with heteroscedastic noise (Appendix A), so on planted structure the TFMs' edge over BART is larger for interactions than for smooth structure, and BART has the edge on step functions with heteroscedastic noise.

Table 2: The TFMs recover a larger share of the removed structure than each comparator. *Share recovered*: the learners' summed rise in test loss when the structure is removed, over its summed worth to the linear fit. *TFMs found more*: tables on which the TFMs' rise exceeds the comparator's, with a sign test. *Mean gap*: the TFMs' extra rise, mean over tables with a 95% BCa bootstrap interval. Top: against BART on the 36 spare-row and 16 new CTR23 tables together and separately, and on the spare-row tables with the seven other TFMs in place of the four or with the structure estimated by TabPFN 3. Bottom: other comparators on the spare-row tables. Medians and variants in Table 6.

Comparator	Share recovered TFMs / comp.	TFMs found more	p	Mean gap [95% CI]
BART, both sets	42% / 23%	37 of 52	0.003	0.048 [0.028, 0.075]
BART, spare-row tables	36% / 21%	23 of 36	0.132	0.033 [0.013, 0.069]
BART, new CTR23 tables	53% / 25%	14 of 16	0.004	0.081 [0.044, 0.132]
BART; seven other TFMs	39% / 21%	26 of 36	0.011	0.039 [0.022, 0.077]
BART; TabPFN 3 estimate	44% / 26%	26 of 36	0.011	0.035 [0.018, 0.076]
BART, context-tuned	36% / 25%	24 of 36	0.065	0.024 [0.010, 0.050]
Boosting, CV-tuned	36% / 23%	28 of 36	0.001	0.029 [0.014, 0.051]
LightGBM, Optuna-tuned	36% / 25%	28 of 36	0.001	0.024 [0.010, 0.043]
CatBoost, CV-tuned	36% / 26%	28 of 36	0.001	0.023 [0.008, 0.042]

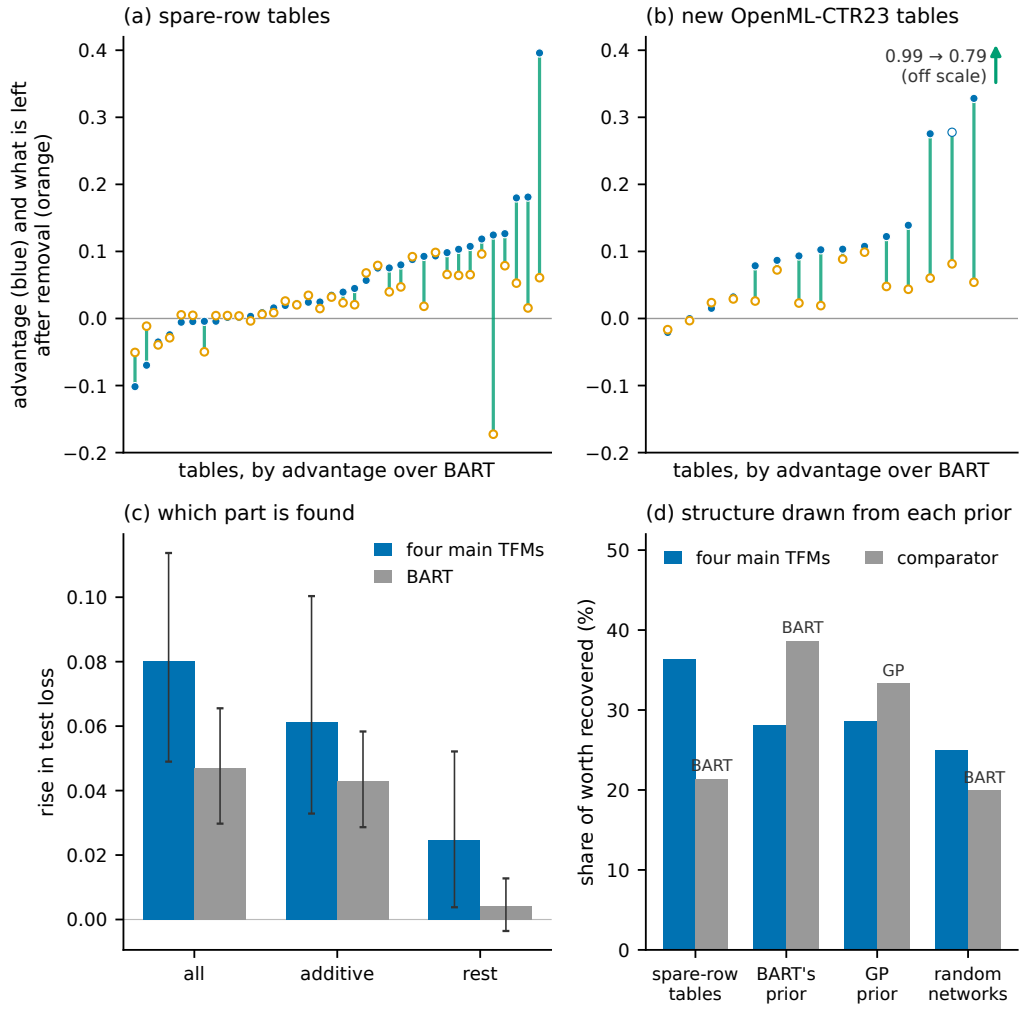

Figure 5: The TFMs recover more of the tables' structure than BART. (a) Spare-row tables, sorted by the TFMs' advantage over BART: the advantage (blue), what is left of it once the structure is removed (orange), and the ablation gap (green). On the point far below zero, removing the structure raises the TFMs' loss but not BART's. (b) The same on the new CTR23 tables; BART's normalized test loss exceeds 1 on the hollow marker's table and on the off-scale one (arrow). (c) Rise in test loss on the spare-row tables when all of the structure, only its fitted additive approximation, or only the rest beyond it is removed (means, 95% intervals). (d) Share recovered, summed over tables, on the spare-row tables and on structure drawn from BART's or the marginalized GP's prior family or from random two-layer networks on the features of all tables (main partition): BART and the GP each recover more than the TFMs on their own prior family.

4.2 Against CV-tuned boosting and LightGBM, the TFMs also fit less noise

The booster's extra noise cost is 20% [12, 30] of the TFMs' advantage over CV-tuned boosting, or 15% when calibrated on simulations in which the redraws overstate a known noise cost (Appendix I). It is 22% against Optuna-tuned LightGBM and small against early-stopped CatBoost (5%).

Averaging CV-tuned boosting's fold models removes most of its noise part. As usually run, it refits one model on all context rows. If its extra noise cost is noise, averaging should remove it and leave the rest. Averaging its five fold models, which cross-validation trains anyway and TabArena's

CV ensembles use (Erickson et al., 2025), cuts its extra noise cost from 20% to 3% of the TFMs' advantage and leaves its ablation gap at 0.027 (0.027 before); the advantage falls by about as much as the noise cost, and the TFMs keep 86% of it (Appendix H).

4.3 The linear-fit part

ΔB is 34% of the advantage over CV-tuned boosting and 38% over BART in Figure 1; over CV-tuned boosting it is about a third on every evaluation set (Appendix D). It compares twin-averaged forecasts, which see the features, the linear fit and the residual's size but not the structure in the residual. Over boosting and BART it arises even without structure beyond the linear fit: when context targets and test labels are drawn from each table's fitted linear model, so that the truth is exactly linear, the trees of boosting and BART, which approximate a linear trend by steps, lose more to least squares than the TFMs do, and the TFMs' ΔB over them is at least as large as on the real tables (Appendix D). Nor does it come from the few test rows where the linear fit extrapolates (Appendix D). ΔB also includes how each learner handles noise in the fitted linear coefficients, which the noise redraws do not count, as they keep the linear fit. Adding the least-squares fit to boosting as an offset does not help: on the main partition the TFMs' advantage over boosting on the least-squares residual is 0.107, against 0.060 over plain CV-tuned boosting (Appendix D).

5 Related work

Statistical ingredients. Twins are the construction behind rotation tests and conditioning on sufficient statistics (Langsrud, 2005; Lindqvist & Taraldsen, 2005), related to co-sufficient sampling (Barber & Janson, 2022) and residual permutation (Freedman & Lane, 1983); residual-prediction goodness-of-fit tests calibrate machine-learning statistics against a linear null with the same scaled-residual draws (Shah & Bühlmann, 2018) and extend to high-dimensional generalized linear nulls (Janková et al., 2020). The noise cost is the variance term of a bias–variance decomposition (Geman et al., 1992), akin to covariance penalties and degrees of freedom (Ye, 1998; Efron, 2004). In-context learners can implement least squares and ridge regression (Garg et al., 2022; Akyürek et al., 2023; von Oswald et al., 2023; Bai et al., 2023). Conditional and black-box randomization tests (Candès et al., 2018; Tansey et al., 2022) and random-label probes (Zhang et al., 2017) are close in spirit; we pair such draws with a structure ablation to locate a black box's advantage.

Tabular foundation models. Müller et al. (2022) frame prior-data fitted networks as approximate Bayesian prediction, Nagler (2023) gives a frequentist bias–variance account of them and Falck et al. (2024) test whether in-context learning in large language models is Bayesian; on real tables we measure the variance side of such an account, the cost of fitting noise, alongside the structure found. Trees versus neural networks (Grinsztajn et al., 2022; McElfresh et al., 2023), what TabPFN v2 learns (Ye et al., 2025), benchmarks (Erickson et al., 2025) and TFMs' internals, predictions, ensembles and fine-tuning (Biloš et al., 2026; Klötergens et al., 2026; Landsgesell et al., 2026; Tanna et al., 2026; Rubachev et al., 2025) are studied elsewhere.

6 Conclusion and limitations

What we find. With 120-row contexts, the TFMs we study find more of our tables' structure than tuned boosters on most tables, and more than BART above all where there is most structure to find, including structure beyond an additive fit that BART barely uses. They also fit less noise than CV-tuned boosting and LightGBM; averaging the former's fold models removes most of the excess but not the ablation gap. About a third of their advantage over CV-tuned boosting is the linear-fit part.

Limitations. Our results use at most 20 features and a common preprocessing. Across context sizes, the net-discovery gap over CV-tuned boosting falls from 0.086 at 60 rows to 0.000 at 1,000 (-0.030 without one outlier-dominated table), while over BART it stays between 0.039 and 0.056 up to 500 rows; against tuned LightGBM the advantage is indistinguishable from zero at 10,000 rows (Appendices L and M). Discovery is measured beyond the linear fit (ten added nonlinear directions change little; Appendix D) and within what the ablation's estimator finds. For 0/1 targets exact twins almost surely do not exist, so we study regression.

AI use statement

In this work, we used generative AI tools to help develop the conceptual framework, propose and refine hypotheses, formulate mathematical claims and help prove them, design and implement the methods and experiments (including data preprocessing and the simulations that generate synthetic datasets), and interpret results. Translation and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to find related literature and to draft and edit the paper. We have reviewed all AI-assisted work: every reported number is generated by scripts from saved result files, the estimators are validated against planted truths in simulation (Appendices A, B and I), each citation was checked against its source, and we checked the proofs and the code. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Ethics statement

This work analyzes publicly available OpenML datasets and involves no human-subjects research or new data collection.

Reproducibility statement

The roster (Table 15), the partitions, the reference basis and the learner settings (Section 3; Appendices B, E, I, J and K) are described in the paper, twins are drawn with fixed seeds, and code and configurations will be released upon publication.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 2019. doi: 10.1145/3292500.3330701.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations*, 2023.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems*, volume 36, pp. 57125–57211, 2023.

Rina Foygel Barber and Lucas Janson. Testing goodness-of-fit and conditional independence with approximate co-sufficient sampling. *The Annals of Statistics*, 50(5):2514–2544, 2022. doi: 10.1214/22-AOS2187.

Marin Biloš, James T. Wilson, Anderson Schneider, and Yuriy Nevmyvaka. A mechanistic study of tabular foundation models. arXiv preprint arXiv:2605.21288, 2026.

Emmanuel Candès, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: 'Model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018. doi: 10.1111/rssb.12265.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010. doi: 10.1214/09-AOAS285.

Bradley Efron. The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99(467):619–632, 2004. doi: 10.1198/016214504000000692.

Moonjung Eo, Min-Kook Suh, Hye-Seung Cho, Jiwon Kim, Seoyoon Kim, Sangjun Nam, and Soonyoung Lee. EXAONE Tabular 1.0: Technical report. arXiv preprint arXiv:2608.25774, 2026.

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. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 38, 2025. doi: 10.52202/085713-0519. URL https://arxiv.org/abs/2506.16791.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Is in-context learning in large language models Bayesian? A martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12784–12805, 2024.

Sebastian Felix Fischer, Liana Harutyunyan, Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – A curated tabular regression benchmarking suite. In *AutoML Conference 2023 (Workshop Track)*, 2023. URL https://openreview.net/forum?id=HebAOoMm94.

David Freedman and David Lane. A nonstochastic interpretation of reported significance levels. *Journal of Business & Economic Statistics*, 1(4):292–298, 1983. doi: 10.1080/07350015.1983.10509354.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598, 2022.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992. doi: 10.1162/neco.1992.4.1.1.

Google Research. TabFM: A zero-shot foundation model for tabular data. https://huggingface.co/google/tabfm-1.0.0-pytorch, 2026. Version 1.0.0.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations*, 2025.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 35, pp. 507–520, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, et al. TabPFN-2.5: Advancing the state of the art in tabular foundation models. arXiv preprint arXiv:2511.08667, 2025.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, et al. TabPFN-3: Technical report. arXiv preprint arXiv:2605.13986, 2026.

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. doi: 10.1038/s41586-024-08328-6.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 26577–26658, 2024. doi: 10.52202/079017-0837.

Rasa Hosseinzadeh, Alex Labach, Zexin Xue, Shuyi Han, Valentin Thomas, and Anthony L. Caterini. TabDPT-Turbo: Efficient in-context learning for tabular prediction. arXiv preprint arXiv:2608.01400, 2026.

Jana Janková, Rajen D. Shah, Peter Bühlmann, and Richard J. Samworth. Goodness-of-fit testing in high dimensional generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3):773–795, 2020. doi: 10.1111/rssb.12371.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Christian Klötergens, Vijaya Krishna Yalavarthi, Lars Schmidt-Thieme, and Tom Hanika. Do tabular foundation models agree with themselves? arXiv preprint arXiv:2608.06004, 2026.

Jonas Landsgesell, Pascal Knoll, and Tizian Wenzel. Distributional regression with tabular foundation models: Evaluating probabilistic predictions via proper scoring rules. arXiv preprint arXiv:2603.08206, 2026.

Øyvind Langsrud. Rotation tests. *Statistics and Computing*, 15(1):53–60, 2005. doi: 10.1007/s11222-005-4789-5.

Bo Henry Lindqvist and Gunnar Taraldsen. Monte Carlo conditioning on a sufficient statistic. *Biometrika*, 92(2):451–464, 2005. doi: 10.1093/biomet/92.2.451.

Antonio R. Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–636, 2018. doi: 10.1080/01621459.2016.1264957.

Si-Yang Liu and Han-Jia Ye. TabSwift: An efficient tabular foundation model with row-wise attention. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2606.07345`.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, pp. 76336–76369, 2023.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022.

Thomas Nagler. Statistical foundations of prior-data fitted networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25660–25676, 2023.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

Prior Labs. TabPFN-2.6. Hugging Face model card, `https://huggingface.co/Prior-Labs/tabpfn_2_6`, 2026.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2602.11139`.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Ivan Rubachev, Akim Kotelnikov, Nikolay Kartashev, and Artem Babenko. On finetuning tabular foundation models. *arXiv preprint arXiv:2506.08982*, 2025.

Rajen D. Shah and Peter Bühlmann. Goodness-of-fit tests for high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):113–135, 2018. doi: 10.1111/rssb.12234.

Synthefy, Po-han Li, Aditya Narayanan, Sai Shankar Narasimhan, Raghav Mallampalli, Aahan Agrawal, Bekzat Ajan, Raimi Shah, and Shubhankar Agarwal. Synthefy Nori: Tabular foundation model for regression. Zenodo, doi: 10.5281/zenodo.20710462, 2026. Nori-30M checkpoint: `https://huggingface.co/Synthefy/Nori-30M`.

Aditya Tanna, Yash Desai, Pratinav Seth, Mohamed Bouadi, Nassim Bouarour, and Vinay Kumar Sankarapu. Ensembling tabular foundation models: A diversity ceiling and a calibration trap. arXiv preprint arXiv:2605.18696, 2026.

Wesley Tansey, Victor Veitch, Haoran Zhang, Raul Rabadan, and David M. Blei. The holdout randomization test for feature selection in black box models. *Journal of Computational and Graphical Statistics*, 31(1):151–162, 2022. doi: 10.1080/10618600.2021.1923520.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174, 2023.

Yuanrui Wang, Xingxuan Zhang, Han Yu, Mingchao Hao, Gang Ren, Hao Yuan, Li Mao, Yunjia Zhang, Chun Yuan, and Peng Cui. LimiX-2M: Mitigating low-rank collapse and attention bottlenecks in tabular foundation models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2606.04485`.

Simon N. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):3–36, 2011. doi: 10.1111/j.1467-9868.2010.00749.x.

Xiaomi-TabLDM Team. Xiaomi-TabLDM: A tabular foundation model technical report. arXiv preprint arXiv:2609.03880, 2026.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at TabPFN v2: Understanding its strengths and extending its capabilities. In *Advances in Neural Information Processing Systems*, volume 38, pp. 135605–135637, 2025. doi: 10.52202/085713-4527.

Jianming Ye. On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131, 1998. doi: 10.1080/01621459.1998.10474094.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Xiyuan Zhang, Danielle C. Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W. Mahoney, Cuixiong Hu, Huzefa Rangwala, George Karypis, and Bernie Wang. Mitra: Mixed synthetic priors for enhancing tabular foundation models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 15795–15840, 2025. doi: 10.52202/085713-0535.

Appendix overview

Every gap Δ in the appendix is signed so that positive favors the TFMs. In order: Appendix A, the twin construction, Propositions 1 and 2, the simulations behind Table 3, other twin laws and an exact per-table test; Appendix B, the structure ablation with its checks (Table 5) and variants (the other TFMs, OpenML-CTR23, native pipelines, other structure estimators and comparators, 250 to 1,000 context rows, known structure, a placebo and a prior-family negative control; Table 6); Appendix C, ranks and the share of the best learner's improvement; Appendix D, the full decomposition, where B comes from and ΔB on a linear truth; Appendix E, all eleven TFMs; Appendix F, the tables outside the models' listed pretraining data; Appendix G, BART variants; Appendix H, averaging the booster; Appendix I, further comparators, native pipelines, TFM ensembles, the noise cost with the structure held fixed and the accounting of Figure 1 (Tables 17 and 18); Appendix J, ensemble and fixed-parameter controls; Appendix K, additional partitions and the evaluation sets of each result (Table 20); Appendices L and M, context sizes from 60 to 10,000 rows; Appendix N, the number of twins and further results.

A Twin construction, further quantities, and Proposition 2 with its proof

Further quantities used in the appendices. A learner's *noise sensitivity*

$$V = \text{mean}_x \, \text{Var}_{k\geq 1} \; p_k(x) \,/\, v \tag{7}$$

is how much its test forecasts move across twins. The *structure gain* $G = \text{mean}_{k\geq 1} L_t(p_k) - L_t(p_0) = D + V$ (with V a variance over the K twins, divisor K) is how much better a learner predicts from the real table than from a typical twin. Since $\bar{p}$ averages K twins, B and D each carry a Monte Carlo term of about $V/(K-1)$ (it cancels in $B - D$), about 0.001 or less at the median V of every TFM, the tuned boosters and BART.

Proposition 1 and its use are standard: it is the conditioning argument behind rotation tests and exact conditional Monte Carlo (Langsrud, 2005; Lindqvist & Taraldsen, 2005), the exact case of cosufficient sampling (Barber & Janson, 2022): under Eq. (1), conditional on $T(y)$, which fixes $P_Z y$ and $\|r\|$, the direction of the residual $(I - P_Z)y$ is uniform on the unit sphere of $\ker(Z^\top)$, and Eq. (2) draws from that law. The argument uses only that the law of ϵ is unchanged by rotations of $\ker(Z^\top)$ that fix $\text{col}(Z)$, which holds for any spherically symmetric noise law, not only the Gaussian. Given $T(y)$, the real target and its twins are therefore independent draws from the same conditional law, hence exchangeable. Under the two conditions below, so are the test losses $L_t(p_0), \ldots, L_t(p_K)$ of any learner; since $\text{mean}_{k\geq 1} L_t(p_k) = L_t(\bar{p}) + V$ exactly, under Eq. (1), $\mathbb{E}\, G = 0$ and $\mathbb{E}\, D = -\mathbb{E}\, V$ for every learner and every K: without structure, expected net discovery is exactly minus the expected noise sensitivity. Rank-deficient designs use the Moore–Penrose projector on the same column space; preprocessing removes exact dependencies before Z is formed.

Two further conditions, implicit in the standard statements, hold for every adapter here: the prediction map is fixed given its context within a run and identical across the twin set (an internally randomized pretrained predictor runs with its seeds, ensemble views and precision fixed before the twin set is formed), and the test and audit rows' inputs and outcomes are independent of the context residual direction under the reference model (they are separate from the context). The propositions also condition on the reference basis, which is selected label-free from context features.

Noise cost and discovery for linear smoothers. A twin turns the whole residual, structure included, into noise, so a noise-sensitive learner loses more on its twins than the structure alone explains. For linear smoothers this is exact:

Proposition 2 (Discovery bracket for linear smoothers) *Let $y = Z\theta + h + e$ with $h \in \ker(Z^\top)$ fixed, $\mathbb{E}\, e = 0$, $\text{Cov}(e) = \sigma^2 I$ and test noise uncorrelated with e, let the learner be a linear smoother $p(x) = w_x^\top y$, and write $m = n - \text{rank}(Z)$. Define the structure share and the learner's true discovery*

$$s = \frac{\|h\|^2}{\|h\|^2 + m\sigma^2}, \qquad \Delta^* = \mathbb{E}\, L_t(\bar{p}) - \mathbb{E}\, L_t(\bar{p} + w_x^\top h).$$

Then, in expectation over e and the twins as $K \to \infty$, with G, D, V and Δ^* unnormalized (multiplied by v),

$$\mathbb{E}\, G = \Delta^* + s\, \mathbb{E}V, \qquad \mathbb{E}\, D = \Delta^* - (1-s)\, \mathbb{E}V.$$

Here Δ^* is the reduction in test loss from the learner's response to the structure h, and $(1-s)\,\mathbb{E}V$ is its expected noise cost, so, in expectation, D is discovery minus noise cost.

Proof of Proposition 2. Write $r = (I - P_Z)y = h + (I - P_Z)e$, $a_x = (I - P_Z)w_x$ and $c = \text{mean}_x \|a_x\|^2$, and take every loss and V unnormalized (multiplied by v), as in the proposition. The reported quantities divide by v, the context variance of the target; with an intercept in Z, v is a function of $T(y)$ and the same for a table and its twins, so the null identity $\mathbb{E}\, D = -\mathbb{E}\, V$ given $T(y)$ holds for the normalized quantities too; the expectations over e below are for the unnormalized quantities, and no corresponding normalized bound is claimed. Because $a_x \in \ker(Z^\top)$, $w_x^\top h = a_x^\top h$ and $w_x^\top (I - P_Z)e = a_x^\top e$. A twin forecast is $p_k(x) = w_x^\top P_Z y + \|r\|\, a_x^\top u_k$ with $\mathbb{E}\, u = 0$ and $\mathbb{E}\, uu^\top = (I - P_Z)/m$, so as $K \to \infty$, $\bar{p}(x) = w_x^\top P_Z y$ and $V = \|r\|^2 c/m$; since $\mathbb{E}\|r\|^2 = \|h\|^2 + m\sigma^2$,

$$\mathbb{E}\, V = \Big(\frac{\|h\|^2}{m} + \sigma^2\Big)c = \frac{\sigma^2 c}{1-s}.$$

The real-table forecast is $p_0(x) = \bar{p}(x) + a_x^\top h + a_x^\top e$. $P_Z e$ and $(I - P_Z)e$ are uncorrelated, $\bar{p}$ depends on e only through $P_Z e$, and the test noise is uncorrelated with the context noise, so the cross term has mean zero and

$$\mathbb{E}\, L_t(p_0) = \mathbb{E}\, L_t(\bar{p} + a_x^\top h) + \sigma^2 c.$$

Hence $\mathbb{E}\, D = \mathbb{E}\, L_t(\bar{p}) - \mathbb{E}\, L_t(p_0) = \Delta^* - \sigma^2 c = \Delta^* - (1-s)\,\mathbb{E}V$, and since $\text{mean}_k\, L_t(p_k) = L_t(\bar{p}) + V$ exactly, $G = D + V$ gives $\mathbb{E}\, G = \Delta^* + s\, \mathbb{E}V$. Of the noise law, only $\mathbb{E}\, e = 0$ and $\text{Cov}(e) = \sigma^2 I$ are needed: $\bar{p}$ is linear in $P_Z e$, so the cross term has mean $\sigma^2 w_x^\top P_Z a_x = 0$ without independence. Both identities, and so the bracket, hold for any homoscedastic noise law whatever its shape; under heteroscedastic noise the noise cost and the cross term both change, and the bracket can fail. For a nonlinear learner, linearizing the forecast in the residual, $p(P_Z y + w) \approx p(P_Z y) + J_x w$ for $w \in \ker(Z^\top)$, gives the same algebra with $a_x = (I - P_Z)J_x^\top$; its accuracy depends on how nonlinear the learner is at the residual's scale. This is a heuristic local approximation: a Jacobian that depends on the fitted noise needs further assumptions, so nonlinear learners do not inherit Proposition 2. In a simulation with planted structure ($n = 120$, four features, $K = 39$, structure shares $s \in \{0.23, 0.54, 0.83\}$), the two identities hold to Monte Carlo error for kernel ridge and distance-weighted k-NN, both linear smoothers; for gradient boosting, a strongly nonlinear learner, the first-order error is 2 to 21% of V. The adaptive learners we audit are checked directly below (Table 3).

The remainder of Eq. (6) for linear smoothers. Write B^- and D^- for B and D with y^- as the real context (losses unnormalized, as in Proposition 2). In the proof ($K \to \infty$), $\bar{p}(x) = w_x^\top P_Z y$, which is the same on y and on the ablated context y^-, since $P_Z y^- = P_Z y$; the least-squares fit and the test rows are also unchanged, so $B^- = B$. When $\hat{f}$ is the true regression function, $(I - P_Z)\hat{f}(X) = h$, so $y^- = y - h$ carries no structure: $\Delta^* = 0$ and $\mathbb{E}\, D^- = -\sigma^2 c$, the negative of the learner's expected noise cost with the structure held fixed (Appendix I). The remainder is the advantage on y^-, which is $\Delta B^- + \Delta D^-$ by Eq. (5), minus ΔB and the extra noise cost, so it is $(\Delta B^- - \Delta B) + (\Delta D^- - \text{extra noise cost})$; both terms are therefore zero in expectation.

Table 3: What holds for the twin quantities. Δ^*: true discovery (Proposition 2).

Setting	What holds
Linear smoothers, homoscedastic noise	$D \le \Delta^* \le G$ in expectation, unnormalized, as $K \to \infty$ (Proposition 2).
Adaptive learners, planted structure	$G \ge \Delta^*$ within Monte Carlo error in all 108 cells, but D is not a lower bound ($D > \Delta^*$ in 53).

Adaptive learners. For the TFMs, BART and the marginalized GP, structure is planted on the real feature matrices of twelve roster tables ($n = 120$; $y = Z\theta + h + e$ with θ the least-squares fit of the real target and h projected onto $\ker(Z^\top)$). The structure is one of six kinds: a smooth function of up to three features, an interaction-heavy function (pairwise and three-way products), a step function with heteroscedastic noise, a high-frequency oscillation, a single sharp threshold on one feature, and the smooth function with the step family's heteroscedastic noise. The noise level gives structure shares $s \in \{0.2, 0.5, 0.8\}$, for 18 settings, with $K = 19$ twins and, per setting, 66 replicates (up to six noise draws per table) for the first three kinds and 33 to 39 (about three per table) for the last three. True discovery is $\Delta^* = L_t(p(Z\theta+e)) - L_t(p(y))$, the loss from the same noise draw with the structure removed minus the loss with it present; for a linear smoother its expectation equals the Δ^* of Proposition 2. The TFMs are the four main TFMs as audited (TabPFN 2.6, TabPFN 3, TabICLv2 and TabDPT-Turbo, one estimator).

G is at or above true discovery in 99 of the 108 learner–setting combinations (six learners times 18 settings) and within Monte Carlo error in 108; every point shortfall is in the smooth structure with heteroscedastic noise at $s \leq 0.5$ (most negative -0.020). D exceeds true discovery in 53 of the 108 learner–setting combinations (Table 4), mostly at high structure share, so for adaptive learners D is not a lower bound. The true ablation gap between a TFM and BART (the gap in Δ^*) depends on the kind of structure: averaged over the four TFMs and three shares it is 0.068 for interaction-heavy structure, 0.015 for smooth structure and -0.016 for step functions with heteroscedastic noise, and the gap in D follows it (0.065, 0.014 and -0.018).

Table 4: Planted-structure simulation with the audited TFMs, BART and the marginalized GP on real feature matrices (66 replicates per row for the first three structures, 33 to 39 for the last three): the smallest amount by which G exceeds true discovery across the six learners and the number of learners with $D > \Delta^*$.

Planted structure	share s	$\min(G - \Delta^*)$ over learners	learners with $D > \Delta^*$
smooth	0.2	0.010	2 of 6
smooth	0.5	0.014	3 of 6
smooth	0.8	0.033	6 of 6
interaction-heavy	0.2	0.007	1 of 6
interaction-heavy	0.5	0.012	2 of 6
interaction-heavy	0.8	0.026	6 of 6
step, heteroscedastic noise	0.2	0.004	2 of 6
step, heteroscedastic noise	0.5	0.007	1 of 6
step, heteroscedastic noise	0.8	0.019	6 of 6
high-frequency oscillation	0.2	0.002	1 of 6
high-frequency oscillation	0.5	0.009	3 of 6
high-frequency oscillation	0.8	0.017	6 of 6
single sharp threshold	0.2	0.001	1 of 6
single sharp threshold	0.5	0.005	2 of 6
single sharp threshold	0.8	0.024	6 of 6
smooth, heteroscedastic noise	0.2	-0.010	0 of 6
smooth, heteroscedastic noise	0.5	-0.020	0 of 6
smooth, heteroscedastic noise	0.8	0.014	5 of 6

Real structure. For each of the 35 tables of Appendix M, 120 of the 2,000 context rows form the context, and their target is replaced by the average of a random forest and gradient boosting fitted to all 2,000 rows plus noise at the fit's out-of-fold residual scale, Gaussian or resampled from the out-of-fold residuals (one draw per table; median structure share 0.34; Δ^* as above). With Gaussian noise the four main TFMs' net-discovery gap over BART, 0.012 [0.0003, 0.025], matches the true gap, 0.013 [0.003, 0.026] (paired difference -0.001 [-0.009, 0.006]). With the tables' own residuals as noise, the gap in D over BART exceeds the true gap by 0.009 [-0.003, 0.022] per table (Hodges–Lehmann; the mean, 0.020 [-0.002, 0.052], is dominated by one table with almost no structure, OpenML 44311). Under both noise laws G is at or above Δ^* within Monte Carlo error for all six learners (the four main TFMs, BART and the marginalized GP).

Table 5: Checks behind the two findings of Section 4, with their results. Planted truths are a tree or a neural network, with Gaussian noise or the table's own; shares are of the removed structure's worth, summed over tables; gaps are the TFMs' extra rise over BART's, with 95% intervals.

Check	Result	Where
Planted tree or network truth	estimated gap within 0.003 of the true one	Tab. 7
Placebo: remove only estimator variance	gap -0.0003 [-0.006, 0.005]	below
Scale of y^-: BART's prior scales frozen	BART's rise changes by -0.001 [-0.003, 0.003]; TFMs standardize targets	below
Structure drawn from each learner's prior family	BART 39% vs TFMs 28%; GP 33% vs TFMs 29%	Fig. 5d
Top five chosen and scored on split halves	advantage 0.182 $\rightarrow$ 0.035	below
Without the time-indexed tables	gap over BART 0.032 [0.006, 0.065]	below
TabPFN 3 and TabICLv2 in native pipelines	gap over BART 0.032 [0.008, 0.062]	below
Partial removal; other variants, BART settings	gap over BART holds	below; App. G
Advantage of each main TFM alone	above zero over CV-tuned boosting and BART	Tab. 18

Twins that keep the residual's own values. A twin draws the residual direction uniformly, which removes, along with any structure, the shape of the real residual: heavy tails, outliers, and noise whose size varies across rows. Two other twin laws keep $T(y)$ exactly and more of that shape. *Permutation twins* permute the real residual; *sign-flip twins* give each row's residual a random sign, so each row keeps its own magnitude before projection and rescaling (the final twin need not); both are projected off col(Z) and rescaled to $\|r\|$. We ran both, and matched Gaussian twins, through the same code on the main partition of all 57 tables ($K = 39$), scoring every law against this run's real-table forecasts. These differ from the main audit's (BART's sampler is random, and the TFMs ran on a different GPU), so the Gaussian values here differ from the main-partition values of Table 16. The four main TFMs' ΔD over CV-tuned boosting is 0.040 [0.019, 0.062] with Gaussian twins, 0.041 [0.017, 0.066] with permutation twins and 0.040 [0.017, 0.064] with sign-flip twins (changes 0.001 [-0.007, 0.009] and -0.0003 [-0.008, 0.008]). Over BART it is 0.033 [0.008, 0.064], 0.034 [0.006, 0.068] and 0.025 [-0.003, 0.057] (changes 0.001 [-0.005, 0.006] and -0.008 [-0.019, 0.001]); per table, the gaps under the other laws correlate with the Gaussian ones at 0.94 or more. The advantage is $\Delta B + \Delta D$ under every law, so what a twin law takes from ΔD it adds to ΔB: keeping each row's residual size before projection moves 0.008 [-0.001, 0.019] of the gap over BART from ΔD to ΔB, a shift whose interval includes zero. The structure ablation does not use twins.

An exact per-table test. Under the reference model the real table and its K twins are exchangeable given $T(y)$ (Proposition 1), for any learner whose prediction map is fixed across the twin set. Let the rank be one plus the number of twins whose loss is at most the real table's; then, with the learner's loss as the test statistic, $p = \text{rank}/(K + 1)$ is a valid p-value against the reference model, exact without ties and conservative with them. We compute it on the 64 audit rows, which are used neither to fit the learners nor to score the test rows ($K = 39$, so the smallest p is 0.025; no twin's loss tied the real table's). At the 5% level the test rejects on 29 to 35 of the 57 tables for the four main TFMs, 27 for BART, 26 for context-tuned BART, 25 for the marginalized GP, 27 for early-stopped CatBoost, 23 for Optuna-tuned LightGBM, 21 for CV-tuned boosting and 6 for CV ridge, while under the reference model the test rejects on each table with probability at most 5%. At least two of the four main TFMs reject on 34 tables; BART does not reject on 9 of these tables, and it rejects on 2 tables where fewer than two of the TFMs do. A rejection indicates that the real residual departs from the reference model in a way the learner can use; the departure can be structure or a residual shape the twins lack, such as heavy tails or noise whose size varies across rows.

B Structure ablation on the real tables

Table 5 lists the checks behind the findings of Section 4, with their results and where each is reported.

For every partition of the 36 re-partitioned tables with at least 1,000 rows outside the partition, we estimate the table's regression function $\hat{f}$ on those rows (re-materialized from OpenML with the same

seed, filtering and preprocessing) in two ways: the average of a random forest (300 trees, minimum leaf size 5) and gradient boosting (300 iterations, learning rate 0.05) on up to 5,000 rows, the estimator of the real-structure simulation (Appendix A), and TabPFN 3 on up to 3,000 rows. We remove the part of $\hat{f}$ beyond the linear fit from the 120 context targets, $y^- = y - (I - P_Z)\hat{f}(X)$, which leaves $Z^\top y$, and hence the linear forecast, unchanged and keeps the table's own noise. A learner's *rise* is the increase in its normalized test loss, $L_t(p(y^-)) - L_t(p(y))$ (both normalized by the real context's target variance), as in Appendix A; it is exactly zero for the linear fit. A partition enters for an estimate only if adding that structure to the linear fit lowers the linear fit's loss on the partition's 64 audit rows (not test rows). The screen keeps 275 of 357 partitions for the tree estimate and 316 of 357 for TabPFN 3; without it the gaps over BART are 0.048 and 0.041. Averaged per table over the kept partitions (7.6 on average), adding the tree estimate's structure lowers the linear fit's test loss by 0.221 [0.112, 0.379], its *worth* to the linear fit.

Table 8 gives the TFMs' ablation gap over each comparator with both estimates, and each learner's rise as a share of the estimate's worth.

BART as the structure estimator. With the structure estimated by BART (defaults, up to 2,000 rows outside the partition; 36 tables, 282 screened partitions), the TFMs' gap is 0.028 [0.009, 0.055] over BART (Hodges–Lehmann 0.009), 0.023 [0.010, 0.038] over context-tuned BART and 0.022 [0.006, 0.046] over CV-tuned boosting: even when BART, fitted on many rows, defines the structure, removing it from 120 rows costs the TFMs more than BART. Its worth to the linear fit on the test rows is small and uncertain on average, and we report no shares of it.

Native pipelines. For TabPFN 3 and TabICLv2 in their native pipelines (Appendix I), against BART, context-tuned BART and Optuna-tuned LightGBM on the same raw features and screened partitions, the gap is 0.032 [0.008, 0.062] over BART (35 tables; BCa interval 0.011 to 0.070; Hodges–Lehmann 0.017), 0.018 [0.001, 0.037] over context-tuned BART and 0.022 [0.006, 0.039] over Optuna-tuned LightGBM. On five tables BART's default noise-scale estimate fails on the raw one-hot features and is set to the target's standard deviation.

A common target transform. Here one Yeo–Johnson transform of the target, fitted on each partition's 120 context targets, is applied to every row for every learner in the native-pipeline ablation; structure and losses are on the transformed scale. We exclude OpenML 44311, where the fitted transform sends the targets outside the split to extreme values and every learner's ablated loss explodes on two partitions (normalized losses above 90 and 1.8). Over the other 34 tables (33 against context-tuned BART) the gap is 0.041 [0.021, 0.078] over BART (BCa; 53% [33, 70] of the advantage), median 0.017 [0.003, 0.040], positive on 82% of tables (sign test $p < 0.001$); over context-tuned BART 0.015 [0.006, 0.025], median 0.013 ($p = 0.035$), positive on 70%; and over Optuna-tuned LightGBM 0.015 [0.004, 0.030], median 0.009 ($p = 0.009$). The gap is more consistent across tables on this scale than on the raw one.

At 250 and 500 context rows. For the four main TFMs and BART (single estimators, the audit's preprocessing, tree estimate, audit-row screen, at least 1,000 spare rows) on all ten partitions of the 36 spare-row tables, 33 tables and 261 partitions pass at 250 rows and 31 and 250 at 500. The gap over BART is 0.023 [0.012, 0.041] at 250 rows (BCa; 39% of the advantage; median 0.008) and 0.017 [0.006, 0.033] at 500 (34%; median 0.002), against 0.033 at 120. BART recovers more of the structure as the context grows. On the tables where it is worth anything, the median shares of the TFMs and BART are 56% and 49% at 250 rows (TFMs more on 20 of 29, $p = 0.061$), 67% and 62% at 500 (19 of 26, $p = 0.029$) and 55% and 26% at 120; summed over tables they are 59% and 45%, 62% and 53%, and 36% and 21%.

At 1,000 context rows. In the native-pipeline ablation on the main partition with 1,000 context rows (30 tables with at least 1,000 further rows outside the partition; no screen), the gap over context-tuned BART is 0.012 [−0.006, 0.029] (median 0.006, positive on 70% of tables, sign test $p = 0.043$), and over Optuna-tuned LightGBM 0.002 [−0.012, 0.016] (median −0.001, positive on 50%): the ablation gap over tuned LightGBM is not detected at 1,000 rows, although the advantage over it is indistinguishable from zero only at 10,000 rows (Appendix M).

Keeping $T(y)$. Removal also changes $y^\top y$. Rescaling the ablated residual to the real one's norm, $\tilde{y} = P_Z y + \|r\|\,(r - h)/\|r - h\|$ with $h = (I - P_Z)\hat{f}$, keeps $T(y)$ exactly, as the twins do. On the same screened partitions (36 tables) the four main TFMs' rise then exceeds BART's by 0.022 [0.006, 0.048] (BCa; median 0.003, 22 of 36 tables, $p = 0.243$) and context-tuned BART's by 0.014

[0.0004, 0.032] (median 0.001, 19 tables, $p = 0.868$); summed over tables the TFMs recover 33% of the structure's worth, BART 23% and context-tuned BART 27%. The mean gap is about two thirds of the removal version's and stays positive; per table it is less consistent.

Keeping the residual's size. Replacing the removed structure by noise of the same norm in $\ker(Z^\top)$ gives a gap of 0.024 [0.004, 0.049] over BART and 0.016 [−0.001, 0.032] over context-tuned BART (36 tables; −0.009 [−0.022, −0.001] from the removal version over BART). Relative to removal, the larger residual raises BART's loss by 0.013 and the TFMs' by 0.004, so this version is not a cleaner estimate of discovery; it keeps the gap over BART positive.

OpenML-CTR23. The same ablation on partitions 0–9 of every OpenML-CTR23 table with at least 1,000 rows outside the partition covers 25 tables and 212 screened partitions (four main TFMs, BART, context-tuned BART). Nine of these tables are other uploads of data in our roster (red wine, diamonds, Miami housing, CPS 1988 wages, Brazilian houses, space_ga, California housing, physicochemical protein, and computer activity, of which cpu_small is a 12-feature version); on the other 16 the gap over BART is 0.081 [0.044, 0.132] (median 0.061, positive on 88% of tables, sign test $p = 0.004$) and over context-tuned BART 0.077 [0.041, 0.133] (median 0.030, 94%, $p = 0.001$); on all 25 it is 0.053 [0.026, 0.091] and 0.049 (Table 6). The three TFMs pretrained only on synthetic tasks (without TabDPT-Turbo) lose more than BART on 14 of the 16 new tables ($p = 0.004$; mean gap 0.089), so the replication does not rest on real-table pretraining.

The seven other TFMs. On the same screened partitions and contexts as in the paragraph *Which part of the structure* below (36 tables; BART refitted alongside), the seven other TFMs (released checkpoints, settings of Appendix E) have a mean gap of 0.039 [0.022, 0.077] over BART (BCa; median 0.012 [0.001, 0.036], positive on 72% of tables, sign test $p = 0.011$; 64% of their advantage) and 0.031 [0.018, 0.055] over context-tuned BART (median 0.011, positive on 81%, $p < 0.001$). Each of these models' mean rise in test loss, 0.066 to 0.106, exceeds BART's 0.047. On the 27 tables where the structure is worth anything to the linear fit, the seven other TFMs recover a median 48% of its worth against BART's 26% (more on 21 of them, sign test $p = 0.006$); the four main TFMs recover a median 55% (21 of 27, $p = 0.006$).

Robust summaries. Per table (Hodges–Lehmann estimate with a table-bootstrap interval, 2,000 resamples; signed-rank test), the four main TFMs' ablation gap is 0.015 [0.002, 0.029] over BART ($p = 0.018$, positive on 64% of tables), 0.013 [0.003, 0.033] over context-tuned BART ($p = 0.005$) and 0.017 [0.003, 0.036] over DART ($p = 0.012$) with the tree estimate, and 0.019 [0.010, 0.036] over BART ($p < 0.001$) with the TabPFN 3 estimate. With BCa intervals (5,000 resamples) the mean gaps are 0.033 [0.013, 0.069], 0.024 [0.010, 0.050] and 0.033 [0.016, 0.060], and their shares of the advantage (ratios of means) 52% [28, 91], 49% [24, 105] and 41% [21, 73]; the ratio of the per-table Hodges–Lehmann estimates of the gap and of the advantage is 27%, 26% and 21%. Per-table medians (order-statistic intervals, exact sign tests) are 0.006 [−0.003, 0.033] over BART ($p = 0.132$), 0.007 [−0.0002, 0.025] over context-tuned BART ($p = 0.065$), 0.006 [0.0004, 0.034] over DART ($p = 0.029$) and, with the TabPFN 3 estimate, 0.016 [0.002, 0.036] over BART ($p = 0.011$); for the rest beyond the additive fit over BART the median is 0.009 [0.0001, 0.018] ($p = 0.029$). In the paragraph *Which part of the structure*, the per-table contrast between the rest beyond the additive fit and the additive part is 0.011 [0.002, 0.021] over BART ($p = 0.018$; mean 0.002 [−0.037, 0.032]) and 0.014 [0.004, 0.028] over context-tuned BART ($p = 0.007$).

Per-table shares and subsets. On the 27 tables where the structure's worth on the test rows is positive, the median share recovered is 55% for the four main TFMs and 26% for BART, and the TFMs recover more on 21 of them ($p = 0.006$; against context-tuned BART 22, DART 22 and CV-tuned boosting 24 of 27); on the 25 where it is worth at least 0.01 of the context variance, 20 ($p = 0.004$); on the new OpenML-CTR23 tables 14 of 15 ($p < 0.001$), and pooled 35 of 42. This subset is selected with the test rows; Table 2 uses all tables. The five largest ablation gaps over BART (*CERN-Electron-Collision-Data*, *methane_emissions_rice_crop*, *seoul_bike_sharing_demand*, *black_friday* and *debutanizer*; one differs from the five largest advantages of Figure 5a) carry 84% of the summed gap; all five are among the seven tables whose structure is worth most, only one is time-indexed, and leaving out any one keeps the mean gap at 0.024 or more (without all five, 0.006). Four are among the 34 drawn tables, which exclude every dataset a model lists in its pretraining data (Appendix F); on the fifth, *CERN-Electron-Collision-Data*, the three TFMs pretrained only on synthetic tasks lose 0.32 to 0.39 of normalized MSE when the structure is removed (TabDPT-Turbo 0.45, BART 0.05), so the gap there does not come from real-table pretraining either. Across tables the gap and the worth

Table 6: The four main TFMs' ablation gap over BART, context-tuned BART and DART, all summaries, and that of the seven other TFMs where marked. Ablation: the TFMs' extra rise in test loss when structure estimated on spare rows (tree estimate) is removed from the context, on the 36 tables with such rows. Native pipelines: TabPFN 3 and TabICLv2 with their own preprocessing and default ensembles against BART on the same raw features, also with one Yeo–Johnson transform of the target, fitted on the context, for every learner (Appendix B). Mean: over tables of each table's partition average, with a 95% bootstrap interval (BCa; percentile for the additive part and the rest); median: over tables, with a distribution-free interval; share: gap over advantage, the advantage averaged over all partitions of these tables (Figure 1 uses the screened partitions beyond the main one). Sign tests and Hodges–Lehmann estimates are in Appendix B.

		BART	BART, context-tuned	DART
Ablation, 36 tables	Advantage, mean	0.064	0.049	0.080
	Gap, mean	**0.033** [0.013, 0.069]	**0.024** [0.010, 0.050]	**0.033** [0.016, 0.060]
	Gap, median; tables positive	0.006 [−0.003, 0.033]; 64%	0.007 [−0.0002, 0.025]; 67%	0.006 [0.0004, 0.034]; 69%
	Gap, mean without the five largest	0.006	0.007	0.010
	Share of advantage, mean	52% [28, 91]	49% [24, 105]	41% [21, 73]
	Gap, mean, seven other TFMs	0.039 [0.022, 0.077]	0.031 [0.018, 0.055]	–
	median; tables positive	0.012 [0.001, 0.036]; 72%	0.011 [0.003, 0.026]; 81%	–
	Gap, mean, structure from TabPFN 3	0.035 [0.018, 0.076]	0.025 [0.014, 0.043]	0.036 [0.018, 0.064]
	median	0.016 [0.002, 0.036]	0.010 [0.001, 0.032]	0.018 [0.001, 0.037]
	Beyond the additive fit only: mean	0.020 [0.002, 0.044]	0.011 [−0.002, 0.024]	–
	median; tables positive	0.009 [0.0001, 0.018]; 69%	0.008 [0.003, 0.016]; 72%	–
	Additive part only: mean	0.018 [−0.004, 0.051]	−0.027 [−0.114, 0.033]	–
	median; tables positive	0.002 [−0.006, 0.005]; 53%	−0.005 [−0.011, 0.002]; 39%	–
Ablation, native pipelines	Gap, mean	0.032 [0.011, 0.070]	0.018 [0.002, 0.037]	–
	Gap, median	0.008 [0.002, 0.031]	0.005 [0.0001, 0.024]	–
Same, Yeo–Johnson targets	Gap, mean	0.041 [0.021, 0.078]	0.015 [0.006, 0.025]	–
	Gap, median; tables positive	0.017 [0.003, 0.040]; 82%	0.013 [0.001, 0.017]; 70%	–
Ablation, OpenML-CTR23, 25 tables	Advantage, mean	0.122	0.097	–
	Gap, mean	0.053 [0.026, 0.091]	0.049 [0.023, 0.090]	–
	Gap, median; tables positive	0.015 [0.003, 0.070]; 72%	0.013 [0.005, 0.041]; 80%	–
	Share of advantage, mean	43%	50%	–
	16 tables not in our roster: gap, mean	0.081 [0.044, 0.132]	0.077 [0.041, 0.133]	–
	median; tables positive	0.061 [0.003, 0.196]; 88%	0.030 [0.005, 0.165]; 94%	–
	share of advantage	47%	59%	–

have rank correlation 0.71. Of the 36 tables, six are time-indexed (rows are consecutive observations of one series); without them the gap is 0.032 [0.006, 0.065] over BART and 0.026 [0.006, 0.050] over context-tuned BART, and the TFMs recover more than BART on 81% of the tables where the structure is worth anything ($p = 0.007$).

Table 2 in full. Summed over tables, the share of the removed structure's worth recovered is 36% for the four main TFMs against 21% for BART, 25% for context-tuned BART, 22% for DART, 23% for CV-tuned boosting (the TFMs' share exceeds BART's by 15 points [7, 25], bootstrap over tables), 25% for Optuna-tuned LightGBM and 26% for CatBoost tuned by 5-fold CV over depth (4, 6, 8), learning rate (0.03, 0.1), L2 leaf penalty (1, 3, 10) and the number of trees (50 to 1,000, chosen on the same folds); the last two use the 36 tables' screened partitions, features and structure estimate. The TFMs lose more than each on 23, 24, 25, 28, 28 and 28 of the 36 tables ($p = 0.132$, 0.065, 0.029, 0.001, 0.001 and 0.001), with mean gaps 0.033, 0.024, 0.033, 0.029 [0.014, 0.051], 0.024 [0.010, 0.043] and 0.023 [0.008, 0.042] (BCa). Against BART, the seven other TFMs recover 39% (BART 21%) and lose more on 26 tables ($p = 0.011$); with the structure estimated by TabPFN 3 the shares are 44% and 26% (26 tables, $p = 0.011$); on the 16 new OpenML-CTR23 tables (worth from the same spare-row fits), 53% and 25% (14 tables, $p = 0.004$); and pooled over the 52 tables, 42% and 23% (37 tables, $p = 0.003$; mean gap 0.048 [0.028, 0.075]).

The advantage and what is left of it. On the screened partitions, the four main TFMs' advantage over BART is 0.202 [0.136, 0.299] on the five tables where it is largest (*CERN-Electron-Collision-Data*, *seoul_bike_sharing_demand*, *black_friday*, *houses* and *methane_emissions_rice_crop*; selected on the advantage) and 0.007 [−0.088, 0.066] on the ablated contexts; on the other 31 tables it is 0.032 [0.013, 0.050] and 0.024 [0.011, 0.038] after removal, positive on 25 of them. Choosing the five tables on a random half of each table's partitions and measuring both quantities on the other half (34 tables with at least two partitions, means over 200 random splits) gives 0.182 and 0.035 on the five and 0.037 and 0.020 on the rest. On the 30 of these 31 tables with the twin audit also run on the ablated contexts (main and first three additional partitions), the advantage is 0.029 [0.004, 0.052], of which the ablation gap is 0.003 [−0.007, 0.012], ΔB on y^- 0.014 [−0.002, 0.027] and ΔD on y^- 0.012 [−0.001, 0.025], adding up exactly per partition; BART's extra noise cost from the noise redraws on the same partitions is −0.005 [−0.010, 0.001].

The TFM side. The four main TFMs are also run on y^- rescaled to the mean and standard deviation of y, with their forecasts mapped back to the scale of y^-. On the 275 screened partitions of the 36 tables their mean rise changes by less than 0.00004 on every table (at most 0.0009 on any partition for any of the four), since each standardizes its targets before prediction.

BART with its prior scales frozen. BART (dbarts at its defaults) sets the scale of its noise prior from the least-squares residual of its targets and rescales the targets to $[-0.5, 0.5]$ by their range before placing its leaf prior. On y^- we refit it with the noise-scale estimate of the real targets y and with its leaf-shrinkage parameter k multiplied by the ratio of the ranges of y^- and y, so the noise prior and the leaf prior's scale are those of the real context (the leaf prior stays centered at the midrange of y^-); passing the default noise-scale estimate explicitly reproduces BART at its defaults up to printing precision. On the 275 screened partitions of our 36 tables, BART's mean rise changes by −0.001 [−0.003, 0.003]; the four main TFMs recover 36% of the removed structure's worth and frozen BART 21%, they lose more on 27 tables ($p = 0.004$), and the mean gap is 0.034 [0.013, 0.068] (BCa; median 0.007). Appendix G gives the gap over other BART configurations.

Which part of the structure. On the same screened partitions (36 tables, 275 partitions), $\hat{a}$ is the fitted additive approximation to the tree estimate $\hat{f}$: gradient boosting with no feature interactions fitted to $\hat{f}$ on the rows outside the partition (median $R^2 = 0.82$ there, linear trend included). With h and h_a the parts of $\hat{f}$ and $\hat{a}$ beyond the linear fit on the context rows, each learner forecasts from $y - h$ (all structure removed), $y - h_a$ (the additive part) and $y - (h - h_a)$ (the rest beyond the additive fit, which holds what interactions add and any additive structure $\hat{a}$ misses). The four main TFMs' gap over BART in the rise in test loss (percentile intervals in this paragraph) is 0.033 [0.010, 0.061] for all structure, as in Table 8; 0.018 [−0.004, 0.051] (Hodges–Lehmann 0.001; positive on 53% of tables) for the additive part, where the TFMs' rise is 0.061 and BART's 0.043; 0.020 [0.002, 0.044] (Hodges–Lehmann 0.011; 69%) for the rest beyond the additive fit, where the TFMs' rise is 0.025 [0.004, 0.052] and BART's 0.004 [−0.004, 0.013]. Over context-tuned BART the gaps are −0.027 [−0.114, 0.033] (Hodges–Lehmann −0.005) for the additive part and 0.011 [−0.002, 0.024] (Hodges–Lehmann 0.009; positive on 72% of tables) for the rest beyond the additive fit: context-

tuned BART uses the additive part at least as well as the TFMs, and the TFMs' edge over it is in the rest. The removals interact, so the parts' gaps need not add up to the gap for all structure.

The twin audit on the ablated contexts. On the first three additional partitions that pass the screen, whose twin audit of y uses the same learner seeds (33 tables, 78 partitions), the four main TFMs, BART and context-tuned BART are run on y^- and its 39 exact twins (the audit's twin seeds). The TFMs' net-discovery gap ΔD over BART is 0.032 [0.013, 0.057] on the real contexts and 0.015 [0.004, 0.027] on the ablated ones; the fall, 0.017 [−0.001, 0.041], tracks the ablation gap on the same partitions, 0.022 [0.003, 0.048] (paired difference −0.005 [−0.008, −0.002], rank correlation 0.92). Over context-tuned BART the gap falls from 0.026 to 0.008, by 0.017 [−0.0001, 0.042] against an ablation gap of 0.022 (rank correlation 0.96). The advantage splits as $\Delta B + \Delta D$ on both contexts, so the paired difference is the change in ΔB: removing the structure changes the TFMs' linear-fit advantage over BART by −0.005 [−0.008, −0.002], although the twins of y^- differ from those of y (the residual norm falls). With ΔB^- and ΔD^- the twin audit's ΔB and ΔD on y^-, the ablation gap is exactly the advantage on y minus the advantage on y^-, so the remainder of Eq. (6) is $(\Delta B^- - \Delta B) + (\Delta D^- - \text{extra noise cost})$; this paired difference is its first part, $\Delta B^- - \Delta B$. On the same partitions (78 partitions of 33 tables), with the noise redraws of CV-tuned boosting and BART, the second part, ΔD^- minus the comparator's extra noise cost, is 0.009 [−0.003, 0.022] against CV-tuned boosting and 0.021 [0.009, 0.033] against BART.

Validation against a known truth. Each partition's tree estimate $\hat{f}$ is planted as the true regression function on its real context, test and spare features, with Gaussian noise at the real context residual's level around $\hat{f}$; it is re-estimated on the synthetic spare rows with the same estimator, and either the true structure or the re-estimated one is removed from the context (36 tables, 275 screened partitions). The TFMs' gap over BART is 0.011 [0.0001, 0.029] with the true structure removed and 0.011 [0.0002, 0.028] with the estimated one (paired difference −0.0002 [−0.002, 0.001]; per-table correlation 0.99, rank correlation 0.92, same sign on 92% of tables); over context-tuned BART 0.008 and 0.008 (difference 0.0003 [−0.002, 0.004], correlation 0.95). Estimating the structure adds no detectable bias to the gap.

Other truths and the table's own noise. On the main and first three additional screened partitions (108 partitions of 35 tables), the planted truth is either the tree estimate or a neural network (two hidden layers of 64 units) fitted to the real spare rows, outside the estimator's family, and the noise is either Gaussian at the real context residual's level or resampled from the real spare rows' residuals around the planted truth; the structure is re-estimated by the tree estimate on the synthetic spare rows in every cell. In all four cells the estimated TFM–BART gap is within 0.003 of the true one (Table 7). The table's own noise raises the true gap, from 0.015 to 0.025 with the tree truth, which accounts for part of the difference from the real-table gap on the same partitions (0.038).

Table 7: Known-truth validation of the structure ablation against BART: the four main TFMs' gap in the rise in test loss with the planted structure removed exactly (true) and with the tree estimate refitted on synthetic spare rows removed (estimated), their difference with a 95% percentile bootstrap interval over tables, and the per-table correlation (108 partitions of 35 tables).

Planted truth	Noise	True gap	Estimated gap	Difference	Correlation
Tree estimate	Gaussian	0.015	0.015	−0.0001 [−0.003, 0.003]	0.98
Tree estimate	table's own	0.025	0.025	0.000 [−0.003, 0.003]	0.99
Neural network	Gaussian	0.016	0.014	−0.003 [−0.007, 0.001]	0.98
Neural network	table's own	0.019	0.020	0.001 [−0.004, 0.007]	0.97

Removing part of the structure. Removing a fraction α of the tree estimate's structure, $y - \alpha(I - P_Z)\hat{f}$, on the same 275 partitions gives a gap over BART of 0.005 [0.0001, 0.012] at $\alpha = 0.25$, 0.016 [0.005, 0.031] at 0.5, 0.023 [0.006, 0.044] at 0.75 and 0.033 [0.010, 0.061] at 1, so the ordering does not depend on how much is removed. Both learners' rises are close to α^2: relative to full removal the TFMs' rise is 0.09, 0.32 and 0.62 at $\alpha = 0.25$, 0.5 and 0.75, and BART's 0.04, 0.20 and 0.56. With a learner's response to the removed structure at the test rows written as $\alpha\delta$, the rise is $\alpha^2\|\delta\|^2$ plus a term linear in α that vanishes when the response is well aimed; a curve close to α^2 thus means the rise at full removal is close to the part of the structure's worth the learner captures, and the share recovered is close to the fraction of that worth it captures.

A prior-family negative control. On the main partition of every table we draw regression functions on the real context and test features from three priors, three draws each, standardize them on the context rows, add standard Gaussian noise, and remove the known structure beyond the linear fit as in the ablation. The prior families are BART's (200 trees; a node splits with probability $0.95(1+\text{depth})^{-2}$, on a uniformly chosen feature at a uniformly chosen observed value; leaf values $\mathcal{N}(0, 1/200)$); the marginalized GP's (an isotropic RBF kernel on the standardized features, with one fixed length scale from that GP's grid, $0.3 \cdot 100^{3/11}\sqrt{d} \approx 1.05\sqrt{d}$); and random two-layer networks on the standardized features (clipped at 5 standard deviations; width 32, weights $\mathcal{N}(0, s^2/\text{fan-in})$ with $s \sim \mathcal{U}(1, 3)$ per layer, standard normal biases, tanh or ReLU activation per layer, output weights $\mathcal{N}(0, 1/32)$). Random networks are one of the function families in TFM pretraining priors (Müller et al., 2022), and differ from both BART's and the GP's prior families. The BART and GP draws follow each learner's prior family rather than its exact fitted prior: they are standardized on the context rows and carry Gaussian noise of fixed unit variance, and the GP draws use one fixed length scale. Under BART's prior family, on the 57 tables where the structure is worth anything (171 draws), BART recovers a median 34% of its worth and the four main TFMs 24%; the TFMs recover more on 6 tables and the mean gap is -0.022 [-0.026, -0.017] (BCa); summed over tables BART recovers 39% and the TFMs 28%. Under the GP's prior family the marginalized GP recovers 29% and the TFMs 25%, the TFMs recover more on 14 of 57 tables, and the mean gap is -0.010 [-0.015, -0.002] (summed: 33% and 29%); on these GP draws the TFMs still recover more than BART on 37 of 57 tables. In each case the learner whose prior family generated the structure recovers the most of it, so the ablation tracks the match between a learner's prior family and the data rather than favoring the TFMs. Under the random-network prior, summed over tables the TFMs recover 25% and BART 20%, the TFMs recover more on 39 of 57 tables ($p = 0.008$), and the mean gap is 0.008 [0.005, 0.013]; the marginalized GP recovers 24% (the TFMs recover more on 33 of 57, $p = 0.289$). With noise resampled from each table's own standardized least-squares residuals instead of Gaussian noise, BART still recovers the most under its prior family (43% against the TFMs' 30%; the TFMs recover more on 5 of 56 tables), and the TFMs still recover more than BART under the random-network prior (24% against 20%; 37 of 57 tables, $p = 0.033$).

Placebo. Two estimates of each kind, $\hat{f}_1$ and $\hat{f}_2$, are fitted on disjoint halves of the same outside rows, and $(I - P_Z)(\hat{f}_1 - \hat{f}_2)/\sqrt{2}$ is removed on the same partitions (36 tables); it contains no structure and has the size of one half-sample estimate's error, larger than the full estimate's. The TFMs' rise in loss differs from BART's by -0.0003 [-0.006, 0.005] with tree estimates and 0.002 [-0.0003, 0.005] with TabPFN 3 estimates (paired over tables, the tree-estimate gap minus its placebo is 0.033 [0.013, 0.058]), so the random part of the estimation error, and a learner's larger response to it, account for almost none of the ablation gap; the placebo does not test a bias the two halves' estimates share. The ablation measures discovery of the structure that $\hat{f}$ captures and can miss structure that neither estimator finds; the tree estimate favors structure that tree ensembles represent well, and the TabPFN 3 estimate structure that TabPFN represents well.

C Ranks and the share of the best learner's improvement

Ranks and normalized scores. Table 9 gives each learner's mean rank of test loss among all 34 learners on each of the 57 tables (main partition, 1 = best), its mean normalized score $(L_{\max} - L)/(L_{\max} - L_{\min})$ per table, with $L_{\max}$ and $L_{\min}$ the highest and lowest test loss among the learners on that table, and for the TFMs the share of tables on which it beats context-tuned BART. Every TFM ranks above every comparator.

The ceiling on each table (Table 10) is the lowest test loss, chosen with the test labels, among 24 candidates: the four main TFMs counted once as a group (through their mean loss) and 23 comparators (every comparator of Table 11, described in Appendices I, J and G). The TFM group is the ceiling on 26% of the tables. A learner's share is its improvement over the linear fit divided by the ceiling's, as a ratio of means over tables with a table-bootstrap interval; the per-table median share is over tables whose ceiling is at least 0.02 below the linear fit. The last column of Table 10 is the paired difference between the TFMs' share and each comparator's, in percentage points. Median normalized losses over tables are 0.550 for the linear fit, 0.455 for BART, 0.364 for the four main TFMs and 0.345 for the ceiling; on 14 of the 57 tables the linear fit is within 0.05 of the ceiling.

Table 8: Structure ablation on the real tables: the four main TFMs' ablation gap over each comparator when structure estimated on rows outside the partition is removed from the context targets (means over the 36 tables, 95% percentile bootstrap intervals over tables; BCa intervals in Tables 6 and 2), next to the twins' net-discovery gap ΔD on the same tables (averages over the same partitions), which also contains the comparator's extra noise cost, and the share of the estimated structure's worth to the linear fit that each learner recovers at 120 rows (ratio of means over tables).

	Ablation gap		Twins
Comparator	Tree estimate	TabPFN 3 estimate	net-discovery gap ΔD
BART	0.033 [0.010, 0.061]	0.035 [0.014, 0.063]	0.041
BART, context-tuned	0.024 [0.008, 0.044]	0.025 [0.013, 0.040]	0.031
DART	0.033 [0.013, 0.056]	0.036 [0.016, 0.059]	0.039
GP, marginalized	0.052 [0.030, 0.077]	0.053 [0.032, 0.076]	0.044
Boosting, CV-tuned	0.029 [0.012, 0.048]	0.034 [0.016, 0.059]	0.048
Share of the estimated structure's worth recovered at 120 rows (%)			
Tree estimate	TFMs 36; BART 21; BART, context-tuned 25; DART 22; GP, marginalized 13; Boosting, CV-tuned 23		
TabPFN 3 estimate	TFMs 44; BART 26; BART, context-tuned 31; DART 26; GP, marginalized 17; Boosting, CV-tuned 27		

Robustness of the shares. Table 11 gives the paired difference between the four main TFMs' share and each of the 23 comparators'. Without the eight time-indexed tables we identified (49 tables), the TFMs capture 76% and BART 57%, a difference of 19 points [7, 36], and 13 [3, 28] against context-tuned BART. With the ceiling learner chosen on each table's 64 audit rows and scored on the test rows, the TFMs beat it on average (share 134%), and the paired differences are 38 points [14, 102] against BART and 26 [7, 79] against context-tuned BART. The seven other TFMs, as a group, capture 76% [64, 86] against the test-label ceiling, 22 points [10, 37] more than BART. With the test-label ceiling, the mean of per-table shares is 70% for the TFMs and 38% for BART. All shares have the same denominator, so a paired difference is the mean advantage divided by the best learner's mean improvement; choosing the ceiling with the test labels favors it, so the test-label shares are conservative for every learner whose improvement is nonnegative.

D Full decomposition at 120 rows

Table 12 gives, for the learners of the main audit other than DART, the ARD GP, the GAM (Appendix G) and the least-squares-offset booster, on the main partition, the median of V and the means of B, D and the net loss at 120 context rows; the TFM row gives the range over the eleven models.

Where B comes from. B is a mean over the test rows, so it splits exactly into the rows on which the least-squares fit extrapolates, those whose leverage $z^\top (Z^\top Z)^+ z$ exceeds the largest leverage among the context rows, and the rest. A median of 2 of the 120 test rows lie beyond the context's leverage range, on 44 of the 57 tables. On these rows the four main TFMs' B is -0.045 of their -0.075, and the boosters and BART gain at least as much over least squares (CV-tuned boosting -0.050, Optuna-tuned LightGBM -0.051, BART -0.048; CV ridge -0.034). The TFMs' ΔB over the boosters and BART therefore comes from the other rows: over CV-tuned boosting it is 0.021 [0.002, 0.042] in all and 0.026 [0.012, 0.044] on the rows within the range; over LightGBM 0.012 and 0.018 [0.009, 0.028]; over BART 0.009 and 0.012 [-0.002, 0.024] (means over tables, 95% bootstrap intervals). Over CV ridge, which gains less on the extrapolating rows, $\Delta B = 0.032$ splits into 0.011 on those rows and 0.022 [0.011, 0.034] on the rest.

A least-squares offset. CV-tuned boosting fitted to the least-squares residual (the offset booster) has $B = -0.0002$, the least-squares value, as it must as $K \to \infty$: the twin residual is sign-symmetric and the booster is odd in its targets, so its twin average is the least-squares forecast. In test loss the offset does not help: on the main partition the TFMs' advantage over the offset booster is 0.107, against 0.060 over plain CV-tuned boosting.

Table 9: Mean rank and normalized score of every learner in the main audit (57 tables, $n = 120$, main partition; 95% bootstrap intervals over tables). ν: the least-squares residual fraction (Appendix J).

Learner	Mean rank	Normalized score	Beats context-tuned BART
EXAONE-Tabular (TFM)	7.8 [6.3, 9.3]	0.93 [0.91, 0.95]	77%
TabPFN 3 (TFM)	8.5 [6.9, 10.2]	0.91 [0.88, 0.94]	77%
Xiaomi-TabLDM (TFM)	9.0 [7.3, 10.8]	0.90 [0.87, 0.93]	77%
Nori-30M (TFM)	9.7 [7.8, 11.7]	0.90 [0.87, 0.93]	75%
TabDPT-Turbo (TFM)	9.7 [7.9, 11.6]	0.90 [0.86, 0.93]	75%
TabPFN 2.6 (TFM)	9.9 [8.3, 11.6]	0.89 [0.85, 0.92]	79%
TabFM 1.0 (TFM)	10.5 [8.4, 12.8]	0.87 [0.82, 0.91]	67%
TabSwift (TFM)	11.0 [9.1, 13.0]	0.88 [0.83, 0.91]	72%
TabICLv2 (TFM)	11.9 [9.9, 14.0]	0.88 [0.85, 0.91]	65%
MITRA v1 (TFM)	11.9 [10.4, 13.5]	0.87 [0.84, 0.90]	72%
LimiX-2M (TFM)	11.9 [10.0, 13.9]	0.88 [0.84, 0.91]	72%
BART, context-tuned	15.4 [13.9, 16.8]	0.82 [0.77, 0.85]	–
BART	16.5 [14.7, 18.2]	0.80 [0.75, 0.84]	–
DART	17.5 [15.3, 19.7]	0.77 [0.72, 0.83]	–
GP, hyperparameters marginalized	17.7 [15.4, 20.0]	0.77 [0.71, 0.81]	–
LightGBM, Optuna-tuned	18.8 [16.7, 20.7]	0.76 [0.71, 0.80]	–
CatBoost, context-tuned	19.1 [16.7, 21.4]	0.75 [0.70, 0.81]	–
Random forest	19.4 [17.2, 21.6]	0.75 [0.71, 0.80]	–
Boosting, CV-tuned	19.5 [17.5, 21.5]	0.74 [0.69, 0.79]	–
GP, fixed kernel, noise var. ν	19.8 [17.6, 21.9]	0.73 [0.68, 0.78]	–
GP (marginal likelihood)	20.0 [17.5, 22.5]	0.67 [0.60, 0.74]	–
Boosting, CV-tuned, least-squares offset	20.0 [18.0, 22.0]	0.72 [0.66, 0.78]	–
GP, ARD	20.1 [17.4, 22.6]	0.68 [0.60, 0.75]	–
Boosting, fixed, 20 bags	21.2 [19.1, 23.1]	0.72 [0.66, 0.77]	–
CatBoost, defaults	21.6 [19.7, 23.6]	0.71 [0.66, 0.76]	–
GAM (REML)	21.8 [19.3, 24.2]	0.65 [0.58, 0.73]	–
Ridge, CV-tuned	21.9 [19.1, 24.5]	0.60 [0.51, 0.69]	–
GP, fixed kernel, noise var. 0.1	21.9 [19.8, 24.0]	0.70 [0.66, 0.75]	–
RealMLP	22.3 [19.9, 24.6]	0.63 [0.54, 0.72]	–
TabM	23.9 [21.9, 25.8]	0.62 [0.54, 0.69]	–
Boosting, fixed	25.0 [23.1, 26.7]	0.62 [0.56, 0.69]	–
MLP, 10-member ensemble	25.5 [23.1, 27.8]	0.51 [0.42, 0.59]	–
k-NN	26.6 [24.3, 28.7]	0.47 [0.38, 0.56]	–
MLP	27.6 [25.2, 29.7]	0.40 [0.30, 0.49]	–

Table 10: Share of the best learner's improvement over the linear fit captured by each learner; 95% table-bootstrap intervals. The ceiling on each table is the lowest test loss among the 23 comparators and the four main TFMs' mean (the TFMs count as one candidate).

Learner	share	median	TFMs − learner
Four main TFMs	76% [63, 87]	85%	–
BART	55% [33, 72]	52%	21 [9, 38]
BART, context-tuned	61% [40, 77]	54%	15 [5, 28]
DART (sparse BART)	62% [35, 77]	54%	14 [−7, 44]
Gaussian process, marginalized	47% [21, 67]	56%	29 [15, 48]
Boosting, CV-tuned	49% [25, 66]	35%	27 [12, 50]
LightGBM, Optuna-tuned	52% [27, 68]	46%	24 [10, 48]
CatBoost, context-tuned	47% [19, 66]	56%	30 [14, 53]

A richer reference model. Proposition 1 holds for any label-free basis with $\text{rank}(Z) \le n - 2$. We also ran the twin audit with Z extended by ten label-free directions built from the squares, cubes and sines of the features (residualized on $[1, X]$ and chosen by their singular values on the context), so that the twins keep this richer fit and D counts only what a learner reads beyond it. The TFMs' gap in net discovery changes little. Over CV-tuned boosting it is 0.039 [0.016, 0.063] with the richer reference and 0.039 with the linear one at 120 rows (57 tables; 0.062 against 0.069 at 60 rows and 0.034 against 0.037 at 240). Over BART, which was run with this basis at 60 and 240 context rows,

Table 11: Paired difference between the four main TFMs' share of the best learner's improvement and each comparator's (percentage points; 95% table-bootstrap intervals; 57 tables).

Comparator	TFMs − comparator	Comparator	TFMs − comparator
DART (sparse BART)	14 [−7, 44]	GP, fixed kernel, noise variance 0.1	49 [30, 77]
BART, context-tuned	15 [5, 28]	RealMLP	61 [32, 113]
BART	21 [9, 38]	Ridge, CV-tuned	61 [39, 92]
LightGBM, Optuna-tuned	24 [10, 48]	Boosting, fixed	65 [41, 110]
Boosting, CV-tuned	27 [12, 50]	TabM	67 [43, 99]
GP, marginalized	29 [15, 48]	GP, ARD, marginal likelihood	68 [38, 117]
Random forest	30 [12, 56]	GP, marginal likelihood	72 [37, 130]
CatBoost, context-tuned	30 [14, 53]	GAM (REML)	83 [61, 113]
GP, fixed kernel, residual-calibrated noise	38 [22, 56]	k-NN	85 [55, 136]
Boosting, fixed, 20 bags	39 [13, 75]	MLP, ten-member ensemble	177 [87, 334]
CatBoost, default	45 [25, 81]	MLP	245 [106, 512]
Boosting, least-squares offset	48 [27, 69]		

it is 0.032 [0.003, 0.072] against 0.035 (60 rows, 58 tables) and 0.025 [0.009, 0.043] against 0.031 (240 rows, 52 tables). Most of the TFMs' edge in net discovery thus lies in what the residual of this richer fit still carries.

The linear-fit share on each evaluation set. ΔB is 34% of the TFMs' advantage over CV-tuned boosting and 38% of that over BART on the partitions of Figure 1, 33% and 34% averaged over all partitions of the re-partitioned tables (Table 21), and 36% and 18% on the main partition alone. The share over CV-tuned boosting is stable across the three sets; over BART, whose ΔB is smaller (0.009 on the main partition), it varies more from one partition to another, and the averages over partitions are the more reliable estimates.

On a linear truth. To see whether ΔB needs structure beyond the linear fit, we rerun the twin audit on each table's own features with the context targets replaced by draws from its fitted linear model, $y^* = Z\hat{\theta} + \hat{\sigma}\epsilon$ with $\epsilon \sim \mathcal{N}(0, I_n)$, and the test labels drawn from the same model (three draws per table, 9 twins, main partition, 57 tables). Nothing beyond the linear fit then exists, in the context or in the test labels. The TFMs' ΔB is 0.042 [0.025, 0.065] over CV-tuned boosting and 0.011 [0.002, 0.022] over BART, against 0.023 [0.004, 0.043] and 0.010 [−0.003, 0.022] on the real tables with the same 9 twins; on the rows within the leverage range it is 0.031 and 0.009. So the TFMs' ΔB over these comparators does not need structure beyond the linear fit: on a linear truth, boosting and BART lose more to least squares ($B = 0.090$ and 0.059) than the TFMs (0.047). Over CV ridge the TFMs' ΔB is −0.013 [−0.030, 0.006] on a linear truth and 0.033 [0.015, 0.055] on the real tables, so their gain over ridge appears only when the test labels carry structure beyond the linear fit. The net-discovery gap on a linear truth is 0.008 [−0.002, 0.018] over CV-tuned boosting and −0.006 [−0.012, 0.0003] over BART.

E Cross-model study: models, inference configurations, and complete results

Models. TabPFN 2.6, TabPFN 3 and TabICLv2 run with one estimator per call through their released regressor interfaces; the point forecast is the mean of 257 requested quantile levels. TabDPT-Turbo 1.2 (real-table pretraining) and the seven other TFMs (TabFM 1.0, TabSwift, MITRA v1, LimiX-2M with its retrieval inference configuration, Xiaomi-TabLDM, EXAONE-Tabular v1 and Nori-30M) use a single inference configuration (float32 inputs, fixed seed) through unchanged adapters; TabDPT-Turbo runs with one ensemble member, and the other seven use eight ensemble views where the release supports them. Six of these eight expose only point predictions, and the audit uses point predictions for all eight. TabDPT-Turbo, the one model pretrained on real tables, has median $V = 0.010$ and mean $B = -0.074$, inside the range of the synthetic-only models (B −0.078 to −0.060). The four main TFMs, all single estimators, span $V = 0.008$–0.025, inside the range 0.007–0.030 of the seven view-averaged models, so view averaging does not drive the comparison. MITRA runs through a resident network that refits only its support-dependent preprocessor per call, which is what its public path reduces to under `fine_tune=False`; the two paths agree at $r = 0.9998$ on a synthetic check.

Table 12: Decomposition at $n = 120$, linear reference model, over 57 tables: median noise sensitivity V (Eq. (7)) with a 95% bootstrap interval, and the means of the linear-fit term B, net discovery D and the net loss against the least-squares fit, $B - D$ (negative is better; exact before rounding). TFM row: range over eleven models (per model in Table 13); middle block: further controls (Appendix J) and comparators (Appendix I). ν: the least-squares residual fraction (Appendix J).

Learner	median [95% CI] noise sensitivity V	means $(L_t(p_0) - L_t^{\text{ls}} = B - D)$ linear-fit B	net discovery D	net loss $B - D$
MLP	0.312 [0.213, 0.445]	0.033	−0.344	0.377
Boosting, fixed	0.138 [0.119, 0.192]	−0.050	−0.025	−0.024
k-NN	0.040 [0.032, 0.049]	0.015	−0.006	0.020
Random forest	0.038 [0.032, 0.049]	−0.053	0.051	−0.104
Boosting, CV-tuned	0.033 [0.026, 0.039]	−0.053	0.057	−0.110
Gaussian process (marginal likelihood)	0.042 [0.020, 0.064]	−0.056	−0.046	−0.010
Ridge, CV-tuned	0.000 [0.000, 0.001]	−0.042	−0.009	−0.033
Least-squares fit (control)	0	0	0	0
MLP, 10-member ensemble	0.269 [0.162, 0.351]	−0.014	−0.240	0.225
Boosting, fixed, 20 bags	0.075 [0.055, 0.095]	−0.054	0.030	−0.084
GP, fixed kernel, noise var. 0.1	0.060 [0.050, 0.081]	−0.038	0.024	−0.062
GP, fixed kernel, noise var. ν	0.010 [0.006, 0.012]	−0.041	0.045	−0.086
CatBoost, defaults	0.106 [0.072, 0.128]	−0.051	0.020	−0.070
CatBoost, context-tuned	0.029 [0.022, 0.035]	−0.035	0.069	−0.104
GP, hyperparameters marginalized	0.007 [0.005, 0.009]	−0.043	0.063	−0.106
BART (dbarts defaults)	0.016 [0.012, 0.021]	−0.066	0.057	−0.123
BART, context-tuned	0.015 [0.010, 0.023]	−0.065	0.072	−0.137
LightGBM, Optuna-tuned	0.027 [0.024, 0.035]	−0.062	0.054	−0.117
RealMLP	0.109 [0.080, 0.123]	−0.061	−0.027	−0.034
TabM	0.039 [0.034, 0.050]	−0.030	−0.011	−0.020
TFMs (range over 11 TFMs)	0.007 to 0.030	−0.078 to −0.060	0.074 to 0.129	−0.207 to −0.144

Roster. The 57 tables (Table 15) are OpenML regression tables that meet these eligibility rules: enough finite-target rows for a 304-row partition of 120 context, 120 test and 64 audit rows, a continuous target with at least 20 distinct values, and at least two nonconstant features after preprocessing. Of these, 34 were drawn at random, balanced over row-count and feature-count strata, from OpenML's 3,081 regression tasks after excluding every dataset a model lists in its pretraining data (TabDPT-Turbo, pretrained on real tables, describes its corpus only by a selection rule over active OpenML datasets, so the drawn tables lie outside the other ten models' pretraining data), known synthetic generators, class-like targets, text-feature tables and duplicate uploads (one upload kept per table name); each drawn table's target is its OpenML regression task's target. The rest are other widely used OpenML regression tables, four of them of classification origin with a numeric column as the target (credit-g: credit amount; diabetes: age; Titanic: fare; ACSIncome: income). Two pairs of tables share a source under different uploads (Conso_RTE and french_load_consumption, both RTE load; datapm2.5 and beijing_pm, both Beijing PM2.5); dropping the later upload of each pair (55 tables) leaves every model's median V unchanged at the reported precision. Results on the 34 drawn tables alone are in Appendix F. Two drawn tables, bank32nh and WorkersCompensation, passed the synthetic-generator filter, but their OpenML descriptions say they are simulated. Without them, removing the structure raises the TFMs' loss more than BART's on 23 of the 34 remaining spare-row tables ($p = 0.058$; mean gap 0.037, against 0.033 with them) and on 37 of 50 together with the new CTR23 tables ($p < 0.001$); both simulated tables have negative gaps.

Complete per-model results. Table 13 gives, for all 11 models, the mean net loss relative to the linear fit and its split into B and D (Eq. (5)), and the median V and D, on the main partition under the linear reference model.

Preprocessing. Each table is preprocessed from its context rows alone, and the same map is applied to the audit and test rows. Numeric columns are imputed with the context median, with a missing-value indicator when a context value is missing. Categorical columns are one-hot coded on their five most frequent context levels plus an *other* level, with the most frequent level as the reference. All columns are standardized, constant columns are dropped, and at most 20 columns are kept by column-pivoted QR on the context design, which uses no labels. Every learner, the TFMs included,

Table 13: Per-model results on the 57 tables (main partition, linear reference model): mean net loss $L_t(p_0) - L_t^{\text{ls}} = B - D$ (negative is better) with its two terms, and median noise sensitivity V and net discovery D. Group: the four main TFMs and the seven others.

TFM	group	net loss (mean)	B (mean)	D (mean)	V (median)	D (median)
TabPFN 2.6	main	−0.182	−0.076	0.106	0.022	0.036
TabPFN 3	main	−0.172	−0.075	0.097	0.008	0.048
TabICLv2	main	−0.160	−0.074	0.087	0.025	0.039
TabDPT-Turbo	main	−0.167	−0.074	0.092	0.010	0.057
TabFM 1.0	other	−0.162	−0.074	0.088	0.030	0.036
TabSwift	other	−0.146	−0.072	0.074	0.009	0.032
MITRA v1	other	−0.173	−0.074	0.099	0.007	0.027
LimiX-2M	other	−0.144	−0.069	0.075	0.029	0.032
Xiaomi-TabLDM	other	−0.181	−0.076	0.105	0.012	0.057
EXAONE-Tabular	other	−0.207	−0.078	0.129	0.007	0.058
Nori-30M	other	−0.182	−0.060	0.122	0.017	0.054

receives the same numeric matrix (no native categorical handling), and the reference basis is built from it.

F Tables outside the models' listed pretraining data

Table 14 gives median noise sensitivity on the 34 drawn tables (Appendix E), which exclude every dataset a model lists in its pretraining data; they lie outside the pretraining data of the ten models pretrained only on synthetic tasks. On these tables median noise sensitivity is 0.006–0.028 against 0.033 [0.026, 0.039] for CV-tuned boosting (equal after rounding to its full-roster median and interval in Table 12), below it for 11 of 11 models and on 71–100% of the drawn tables paired by dataset. For the ten models pretrained only on synthetic tasks, the sensitivity gap does not depend on datasets in their pretraining data.

The drawn tables include the 8 tables (from six sources) we identified whose rows are consecutive observations of one series (stock prices, electricity load, air quality, bike demand, rainfall and river flow: rainfall_bangladesh, datapm2.5, AAPL_stock_price_2021_2022_1, Reading_Hydro, beijing_pm, seoul_bike_sharing_demand, Conso_RTE, french_load_consumption), where random splits make temporal interpolation available to every learner. The sensitivity ordering is unchanged on these tables and on the other 49 (every model below tuned boosting). On the 49 other tables alone, the four main TFMs' advantage splits into linear-fit plus net discovery as $0.057 = 0.020 + 0.038$ against tuned boosting, $0.045 = 0.006 + 0.039$ [0.010, 0.078] against BART and $0.032 = 0.007 + 0.025$ against context-tuned BART; averaged over up to ten partitions of the 47 of them that were re-partitioned, the net-discovery gap is 0.034 [0.017, 0.054] against BART and 0.040 [0.025, 0.055] against DART. The comparison against BART, dominated by net discovery, thus does not depend on the time-indexed tables.

G BART variants

DART (Linero, 2018) is BART with a Dirichlet sparsity prior on the splitting proportions (`wbart` in the R package `BART`, `sparse=TRUE`, 200 trees, 1,000 posterior draws after 100 burn-in). The structure ablation also uses BART with a weaker depth penalty (`dbarts`, power 1 instead of 2) and BART with 1,000 trees instead of 200. For the ranks and the ceilings of Appendix C, DART, an ARD Gaussian process and a GAM with REML smoothness selection (`mgcv`; Wood, 2011) are also fitted to each real table and scored on its 64 audit rows.

A wider tuning grid. Context-tuned BART chooses the leaf shrinkage $k \in \{1, 2, 3\}$ and the number of trees (50 or 200) by 5-fold cross-validation on the context; the wider tuning applies the same cross-validation to BART (`wbart`, 200 trees) over 24 settings (DART's Dirichlet sparsity prior on or off, $k \in \{1, 2, 3\}$, tree-depth prior power in $\{1, 2\}$ with base 0.95, noise prior quantile in $\{0.9, 0.99\}$)

Table 14: Median noise sensitivity V on the 34 drawn tables, which exclude every dataset a model lists in its pretraining data, $n = 120$, linear reference model. Tuned boosting's sensitivity equals its full-roster median after rounding.

Learner	drawn tables (34) V
EXAONE-Tabular	0.008
LimiX-2M	0.028
MITRA v1	0.006
Nori-30M	0.016
TabDPT-Turbo	0.009
TabFM 1.0	0.023
TabSwift	0.009
Xiaomi-TabLDM	0.012
TabICLv2	0.022
TabPFN 2.6	0.020
TabPFN 3	0.009
Boosting, CV-tuned	0.033
Boosting, fixed	0.158

and refits the selected setting with 1,000 posterior draws. On the main and three additional partitions (54 tables with all runs complete), its net loss is -0.078 against -0.086 for the narrower tuning (paired difference 0.009 [-0.013, 0.026]), -0.071 for default BART and -0.056 for DART. It is more noise-sensitive than the four main TFMs ($\Delta V = 0.006$), and their net-discovery gap over it is 0.015 [-0.002, 0.028]; over the narrower tuning it is 0.020 [0.008, 0.031] on the same tables and partitions. Widening the grid does not tune BART further.

Other BART configurations. On the screened partitions of the structure ablation (Appendix B), the four main TFMs' ablation gap over BART with its noise prior centered on a nonparametric residual scale (the out-of-bag residual standard deviation of a 500-tree random forest on the context, instead of the least-squares residual's) is 0.034 [0.011, 0.063] (median 0.007, positive on 69% of tables, sign test $p = 0.029$), against 0.033 over default BART on the same partitions, so the least-squares noise scale does not explain the gap. Over BART tuned on the context over the wider 24-setting grid above, the gap is 0.017 [0.004, 0.030] (median 0.012, positive on 67% of tables, sign test $p = 0.065$). BART with a weaker tree-depth penalty (power 1 instead of 2) recovers a median 27% of the structure's worth and BART with 1,000 trees 19%, against the TFMs' 55%; the TFMs recover more on 20 and 23 of the 27 tables where it is worth anything ($p = 0.019$ and $p < 0.001$), with mean gaps 0.026 [0.009, 0.058] and 0.038 [0.018, 0.079] (BCa), so neither a deeper nor a larger forest closes the gap.

H Averaging the booster

Averaging CV-tuned boosting over the five training parts of a 5-fold split (the fold models of cross-validation, as in the cross-validation ensembles of TabArena; Erickson et al., 2025) lowers its loss by 0.012 [0.006, 0.018] on the 477 partitions where it was run (55 tables). Averaging over 39 bootstrap resamples of the context does not lower its loss (reduction -0.004 [-0.017, 0.009]); averaging over 39 subsamples of 80% of the rows lowers it by 0.024 [0.012, 0.036] (84 partitions, 28 tables).

Averaging as a check of the noise part. If the accounting of Figure 1 reads the noise part right, averaging the booster should remove it and leave its ablation gap. The booster is CV-tuned as before (same grid, folds and seed) and then forecasts with the mean of its five cross-validation fold models at the chosen setting; it is run through the noise redraws and the structure ablation on the 237 partitions of Figure 1 (35 tables). Its extra noise cost over the four main TFMs falls from 0.014 [0.008, 0.020] to 0.002 [-0.003, 0.006] (a fall of 0.012 [0.009, 0.015]; 20% of the advantage before, 3% after). The TFMs' advantage falls from 0.072 to 0.062 (by 0.010 [0.001, 0.019]; real-context losses from the ablation runs); the fall in the advantage minus the fall in the noise cost is -0.002 [-0.010, 0.006]. The ablation gap is 0.027 [0.008, 0.050] before and 0.027 [0.012, 0.046] after.

Table 15: The 57 tables: OpenML data and task identifiers, source rows and raw features, and columns kept after one-hot coding and the label-free 20-column cap. †One of the 34 tables drawn after excluding the models' listed pretraining data (Appendix F); the OpenML descriptions of bank32nh and WorkersCompensation say they are simulated (Appendix E).

OpenML data id	task id	name	target	rows	features	kept
31	146813	credit-g	credit_amount	1000	19	20
37	211838	diabetes	age	768	7	7
227	2315	cpu_small†	usr	8192	12	12
315	4773	us_crime	ViolentCrimesPerPop	1994	122	20
500	4830	analcatdata_vineyard†	Lugs	468	3	7
507	4835	space_ga†	ln(VOTES/POP)	3107	6	6
537	4862	houses	median_house_value	20640	8	8
538	4863	colleges_usnews	Graduation_rate	1204	32	20
543	4868	boston_corrected†	LSTAT	506	19	20
556	4879	analcatdata_apnea2†	Count	475	3	9
558	4881	bank32nh†	rej	8192	32	20
575	4894	kdd_coil_4†	algae_4	316	11	20
673	4990	chscase_census2†	col_8	400	7	7
688	4999	visualizing_soil†	track	8641	4	4
690	5001	visualizing_galaxy	velocity	323	4	4
23516	360051	debutanizer†	y	2394	7	7
40691	211847	wine-quality-red	alcohol	1599	10	10
41265	168890	Titanic	Fare	1307	7	7
41506	168869	NewFuelCar	Tmax	36203	17	18
41514	189764	Diabetes(scikit-learn)	class	442	10	10
41539	168889	rainfall_bangladesh†	Rainfall	16755	3	11
41540	168891	black_friday†	Purchase	166821	9	17
41542	168887	CD4	Future_CD4	16484	61	20
41700	189931	CPMP-2015-regression†	runtime	2108	26	20
42208	233210	nyc-taxi-green-dec-2016	tip_amount	581835	11	18
42225	233211	diamonds†	price	53940	9	20
42236	211247	datapm2.5†	pm2.5	43800	9	9
42363	211853	forest_fires	area	517	12	20
42370	211860	yacht_hydrodynamics	Residuary.resistance	308	6	6
42372	211864	autoMpg	Mpg	392	5	5
42688	317612	Brazilian_houses†	total_(BRL)	10692	12	19
42721	359926	Airlines_DepDelay_1M	DepDelay	1000000	9	20
42876	362305	WorkersCompensation†	UltimateIncurredClaimCost	100000	13	20
42903	360950	physicochemical-protein†	RMSD	45730	9	9
43141	362339	ACSIncome	PINCP	1664500	11	11
43429	362360	CERN-Electron-Collision-Data	M	99915	16	16
43785	362390	Goodreads-Computer-Books	Avg_Rating	1234	6	6
43828	362394	Another-Dataset-on-used-Fiat-500-(1538-rows)	price	1538	7	8
43881	361014	AAPL_stock_price_2021_2022_1†	Close	347	5	5
43926	362417	ames_housing†	Sale_Price	2930	80	20
43963	361030	CPS1988†	wage	28155	6	8
44212	361131	Reading_Hydro†	upstream	1000	26	9
44311	362589	Job_Profitability†	Jobs_Gross_Margin_Percentage	14480	29	20
44994	361270	cars	Price	804	17	14
45043	361289	seattlecrime6†	ReportedTime	52031	4	11
46285	362091	beijing_pm†	pm2.5	41757	11	13
46295	362095	qsar_aquatic_toxicity	8	546	8	8
46297	362096	seoul_bike_sharing_demand†	13	8760	13	19
46337	362125	Conso_RTE†	Load	105168	15	15
46340	363263	french_load_consumption†	Load	105168	16	20
46586	363399	QSAR_Bioconcentration_regression	logBCF	779	9	9
46662	363376	news_popularity2†	log_shares	24007	4	8
46729	363437	climate_insights_dataset†	Sea_Level_Rise	10000	8	20
46736	363444	methane_emissions_rice_crop†	Value	138658	16	13
46747	363447	IoT_Agriculture_2024†	water_level	37922	12	13
46772	363471	munich-rent-index-1999†	rent	3082	8	9
46942	363649	miami_housing†	SALE_PRC	13776	15	14

Table 16: The four main TFMs' net-discovery gap ΔD on the main partition (57 tables) and averaged per table over the main and three additional partitions, and, averaged per table over all partitions (55 tables; ten partitions for 52 of them, four for the rest), the advantage, its linear-fit part ΔB, the sensitivity gap ΔV, ΔD and the net-discovery share (95% table-bootstrap intervals; positive favors the TFMs).

	main partition	four partitions		mean over all partitions			
Comparator	ΔD	ΔD	advantage	ΔB	ΔV	ΔD	ΔD share
BART	0.039 [0.012, 0.072]	0.032 [0.016, 0.052]	0.050 [0.034, 0.070]	0.017 [0.010, 0.025]	−0.0001 [−0.003, 0.003]	0.033 [0.018, 0.052]	66% [48, 80]
BART, context-tuned	0.024 [0.003, 0.049]	0.019 [0.009, 0.032]	0.035 [0.023, 0.048]	0.014 [0.007, 0.021]	0.001 [−0.001, 0.004]	0.021 [0.011, 0.032]	60% [42, 76]
GP, marginalized	0.033 [0.002, 0.067]	0.028 [0.012, 0.044]	0.068 [0.046, 0.092]	0.037 [0.025, 0.050]	−0.010 [−0.012, −0.008]	0.031 [0.017, 0.047]	46% [32, 57]
Boosting, CV-tuned	0.039 [0.016, 0.062]	0.044 [0.031, 0.058]	0.064 [0.048, 0.081]	0.021 [0.013, 0.029]	0.021 [0.017, 0.024]	0.043 [0.031, 0.057]	67% [58, 77]
DART (sparse BART)	0.008 [−0.035, 0.036]	0.032 [0.015, 0.048]	0.066 [0.047, 0.085]	0.030 [0.019, 0.042]	0.005 [0.003, 0.008]	0.036 [0.023, 0.050]	55% [42, 67]

I Additional comparators and robustness checks

Native pipelines. Instead of the audit's preprocessed matrix (at most 20 features, top categorical levels one-hot coded), the four main TFMs receive the table's raw features through their released interfaces, on the main audit's twins (same seeds and row indices) with the same checkpoints: TabPFN 2.6 and TabPFN 3 with every raw feature, categorical features passed as categorical, missing values kept and the default eight-member ensemble; TabICLv2 and TabDPT-Turbo with categorical features coded as integers, missing values kept (TabDPT-Turbo with its missing-indicator option enabled), each package's default ensemble and its native point prediction. BART and DART receive the same raw features (numeric features median-imputed, categorical ones one-hot coded, no feature cap; where the one-hot columns outnumber the rows, BART's noise prior is set from the target's standard deviation). BART did not run on the raw features of one table (OpenML 46736), which the raw-feature comparisons omit. Relative to the audited single-estimator TabPFNs, the native TabPFNs lower their net loss by 0.044 [0.006, 0.092], and BART gains slightly more from the raw features. Against raw-feature BART and DART, the two TabPFNs' advantage is 0.043 [0.011, 0.077] and 0.060 [0.029, 0.091], of which net discovery is 0.028 [0.002, 0.057] and 0.034 [0.010, 0.059] (56 tables). With all four main TFMs native, net loss falls by 0.049 [0.012, 0.095] relative to the audited configuration; against raw-feature BART the advantage is 0.035 [−0.0005, 0.071] and the net-discovery gap 0.024 [−0.001, 0.052] (per-table Hodges–Lehmann 0.019), and against raw-feature DART 0.052 [0.019, 0.085] and 0.030 [0.008, 0.052] (56 tables, main partition). Averaged per table over the main partition and the three additional partitions of Appendix K (54 tables, same row indices), the advantage over raw-feature BART is 0.055 [0.028, 0.087], of which net discovery is 0.033 [0.013, 0.058] (59%; per-table Hodges–Lehmann 0.015, positive on 78% of tables), and over raw-feature DART net discovery is 0.037 [0.020, 0.058].

GP, hyperparameters marginalized is an isotropic RBF-plus-noise Gaussian process on standardized features and targets whose length scale and noise fraction are integrated over a 12×12 grid (12 log-spaced length scales from 0.3 to 30 times $\sqrt{d}$, 12 evenly spaced noise fractions from 0.02 to 0.98) under a uniform prior, with weights proportional to the marginal likelihood; it is the Bayesian analog of the GP with fixed kernel and residual-calibrated noise (Appendix J). *BART* is Bayesian additive regression trees (Chipman et al., 2010) through `dbarts` at its defaults (200 trees, 1,000 posterior draws after 100 burn-in), posterior mean, fixed seed. Its default noise-scale prior is calibrated to the least-squares residual standard deviation (placed at its 0.9 quantile), a function of $T(y)$ and hence identical across the twin set, the calibration the GP with fixed kernel and residual-calibrated noise receives explicitly; its leaf-mean prior is scaled by the target range, which varies across twins. Its Monte Carlo error is small: with 4,000 posterior draws after 500 burn-in on the same twins, its mean D changes by 0.0001 [−0.001, 0.001] per table and its mean V by −0.0003. Across 20 seeds on the real table of each main-partition task (57 tables), the Monte Carlo variance of its forecast is 0.0011 of the target variance on average (median 0.0002); the real-table forecast carries this error in full and the twin average less (a $1/K$ share if the errors are independent across twins; the shared seed may correlate them), so it lowers BART's D, and raises the TFMs' gap, by at most about that amount. *BART, context-tuned* selects the leaf shrinkage $k \in \{1, 2, 3\}$ and the number of trees $\in \{50, 200\}$ by 5-fold cross-validation on the context and refits (for CatBoost, "context-tuned" is a validation-split rule; both use only the twin-set member's context).

At its defaults CatBoost does not reach TFM-level insensitivity: $V = 0.106$ [0.072, 0.128], close to fixed boosting and three times CV-tuned boosting, with mean net loss −0.070. Early-stopped CatBoost reaches the top of the TFM range (0.007–0.030): $V = 0.029$ [0.022, 0.035] and mean net loss −0.104, beside CV-tuned boosting (0.033 and −0.110), level with the two most sensitive TFMs (TabFM 1.0, LimiX-2M) and above the other nine. BART and the marginalized GP reach TFM-level insensitivity: the marginalized GP has $V = 0.007$ [0.005, 0.009] and BART $V = 0.016$ [0.012, 0.021], both inside the TFM range. BART's linear-fit term (−0.066) is close to the TFMs' (−0.078 to −0.060); the marginalized GP's (−0.043) is not ($\Delta B = 0.032$). Their mean net discovery, 0.063 and 0.057, is below the TFMs' 0.074–0.129. In mean net loss BART (−0.123) sits between CV-tuned boosting (−0.110) and the TFMs (−0.207 to −0.144), and the marginalized GP (−0.106) is level with tuned boosting. Tuning BART on the context raises its mean net discovery to 0.072, just below the lowest TFM (0.074), and lowers its net loss to −0.137, the best of the twenty comparators other than DART that run on the same twins (DART captures a slightly larger share of the best learner's improvement on the main partition; Table 10), while leaving $V = 0.015$ (0.016 at defaults),

unlike refitting the GP's hyperparameters, which raises sensitivity. The four main TFMs' advantage over tuned BART is 0.034 [0.012, 0.056] = 0.010 + 0.024 (linear-fit plus net discovery; $\Delta V = 0.001$): tuning narrows the main-partition net-discovery gap without closing it. Posterior averaging under a generic prior, as in BART, thus reproduces the twin side of the TFM account (low across-twin variance and a good estimate of the linear-fit signal), and the net-discovery half separates the pretrained prior from default BART's; a sparsity prior on the splitting proportions does not close it once partition noise is averaged out (Table 16).

Stronger boosting and modern deep baselines. *LightGBM, Optuna-tuned* (Ke et al., 2017; Akiba et al., 2019) selects, on each twin-set member's own context, the number of trees (25 to 1,000), learning rate, number of leaves, minimum leaf size, row and column subsampling and L_2 penalty by 50 trials of a seeded TPE sampler scored by 5-fold cross-validated MSE, then refits on the full context. *RealMLP* at its tuned defaults (RealMLP-TD; Holzmüller et al., 2024) and *TabM* at its default hyperparameters (TabM-D; Gorishniy et al., 2025), both through `pytabkit`, run on CPU with a fixed seed, each holding out a fifth of the context (about 24 rows) for best-epoch selection. Tuned LightGBM is the strongest booster in this study (mean net loss -0.117, against -0.110 for CV-tuned boosting), yet every TFM has lower net loss and higher net discovery (0.054 in mean); the four main TFMs' advantage over it is 0.054 [0.026, 0.083] = 0.012 + 0.041. RealMLP and TabM are more sensitive than every TFM (median $V = 0.109$ and 0.039) with negative mean net discovery (-0.027 and -0.011), and the TFMs' advantage over them is 0.136 and 0.151.

Noise cost with the structure held fixed. The noise cost holds the table's structure fixed and redraws only the noise, on the 237 screened partitions of the structure ablation used in Figure 1 (35 of the 36 tables with at least 1,000 spare rows). With $\hat{f}$ the tree estimate fitted on the spare rows and $e = y - \hat{f}(X)$ the context rows' out-of-sample residual from it, the j-th redraw is $y_j = P_Z y + (I - P_Z)\hat{f}(X) + (I - P_Z)e_{\pi_j}$ for a random permutation π_j of the context rows ($j = 1, \ldots, 9$, the same permutations for every learner), so the linear fit and the structure beyond it are those of the real table and the noise keeps its values, reordered across rows, before projection. A learner's noise cost is the variance of its test forecasts across the redraws (divisor 8), normalized as V; it adds one for one to the learner's expected test loss at this structure. For a linear smoother under homoscedastic noise, when $\hat{f}$ is the true regression function, the expected noise cost (unnormalized) is $\sigma^2 c = (1-s)\,\mathbb{E}V$ (Proposition 2): the intercept in Z gives $1^\top a_x = 0$, so the permutation variance is $\|a_x\|^2$ times the sample variance of e (divisor $n-1$), whose expectation is σ^2. Since Proposition 2 also gives $\mathbb{E}\,D = \Delta^* - (1-s)\,\mathbb{E}V$, under these conditions the gap ΔD is, in expectation, the ablation gap plus the comparator's extra noise cost, so ΔB, the extra noise cost and the ablation gap add up to the advantage in expectation (Figure 1); the remainder measures the departure of adaptive learners and of an estimated $\hat{f}$ from this. Like the twins, the redraws hold $P_Z y$ fixed, so the variance a learner takes from the noise's component in the column space of Z enters the linear-fit term, not the noise cost. The four main TFMs' mean noise cost is 0.025, CV-tuned boosting's 0.039, Optuna-tuned LightGBM's 0.038, early-stopped CatBoost's 0.028 and BART's 0.018. As a share of the TFMs' real-table advantage on the same partitions (ratio of means over tables, percentile bootstrap over tables), the comparator's extra noise cost is 22% [11, 33] for LightGBM and 5% [-1, 12] for CatBoost; for CV-tuned boosting, BART and context-tuned BART, whose twin audits cover the same partitions, Figure 1 and Table 17 give the full accounting.

Checking the noise cost against a known truth. On up to three screened partitions of each table (102 partitions of 36 tables), the tree estimate, refitted on the stored spare rows, is planted as the true function on the context, test and spare features, with Gaussian noise at the real context residual's level, and re-estimated on the synthetic spare rows. The true noise cost is the forecast variance across independent redraws of the noise with the true structure held fixed; the measured one is the variance across permutations of the residual around the re-estimate, as on the real tables. For the four main TFMs the measured noise cost is on average 1.05 times the true one, for CV-tuned boosting 1.16 times: the permuted residual also carries structure the estimate misses, which a noise-sensitive learner partly fits. The booster's extra noise cost over the TFMs is 0.020 [0.015, 0.024] measured against 0.015 [0.011, 0.020] true (difference 0.005 [0.002, 0.007]; 96 partitions of 34 tables), and BART's is -0.008 against -0.007. Scaling the real-table shares by the ratio of the measured to the true gap in these simulations (1.29), a calibration that assumes the real tables share that ratio, gives

15% of the advantage over CV-tuned boosting and, with the same ratio, 17% over LightGBM: the boosters' extra noise cost is 15 to 22% of the TFMs' advantage over them, corrected or as measured.

Table 17: One accounting of the four main TFMs' advantage (Figure 1) on the 237 partitions of 35 of the 36 tables with spare rows: the linear-fit part from the twins (ΔB), the comparator's extra noise cost with the structure held fixed, the ablation gap from the structure ablation, and the remainder. Means over tables with 95% percentile bootstrap intervals; shares of the advantage in parentheses. †On the 236 partitions with its noise redraws.

Comparator	Advantage	Linear-fit (ΔB)	Extra noise cost	Ablation gap	Remainder
Boosting, CV-tuned	0.072 [0.047, 0.100]	0.025 (34%) [0.016, 0.034]	0.014 (20%) [0.008, 0.020]	0.027 (38%) [0.009, 0.049]	0.006 (8%) [−0.004, 0.017]
BART	0.057 [0.033, 0.086]	0.022 (38%) [0.011, 0.032]	−0.006 (−11%) [−0.010, −0.002]	0.026 (46%) [0.007, 0.050]	0.015 (27%) [0.006, 0.025]
BART, context-tuned†	0.042 [0.024, 0.061]	0.018 (43%) [0.009, 0.027]	0.0005 (1%) [−0.005, 0.005]	0.017 (41%) [0.006, 0.031]	0.006 (15%) [−0.004, 0.017]

Table 18: The accounting of Figure 1 for each main TFM alone, on the same partitions: its advantage (95% percentile bootstrap interval over tables) and the shares of it that are linear-fit, the comparator's extra noise cost, the ablation gap and the remainder.

TFM	comparator	advantage	linear-fit	extra noise cost	ablation gap	remainder
TabPFN 2.6	Boosting, CV-tuned	0.070 [0.045, 0.100]	30%	14%	46%	10%
TabPFN 3	Boosting, CV-tuned	0.076 [0.048, 0.106]	40%	24%	33%	2%
TabICLv2	Boosting, CV-tuned	0.064 [0.040, 0.089]	33%	12%	42%	12%
TabDPT-Turbo	Boosting, CV-tuned	0.077 [0.049, 0.108]	34%	26%	31%	9%
TabPFN 2.6	BART	0.056 [0.030, 0.084]	32%	−19%	57%	30%
TabPFN 3	BART	0.061 [0.033, 0.093]	45%	−3%	40%	18%
TabICLv2	BART	0.049 [0.025, 0.076]	37%	−25%	53%	35%
TabDPT-Turbo	BART	0.062 [0.037, 0.093]	37%	0%	37%	26%

An exact split from noise redraws on both contexts. On the partitions of Figure 1 the noise redraws are also run on the ablated context, with the same nine permutations: $y_j^- = P_Z y + (I - P_Z)e_{\pi_j}$. For each learner and context let m be the mean of its nine test forecasts; its mean loss over the redraws is exactly $L_t(m)$ plus its noise cost (here with divisor 9). The TFMs' advantage averaged over the redraws on y therefore splits exactly, with no remainder, into a *linear-fit part*, the comparator's $L_t(m)$ on y^-, the context where no estimated structure is left, minus the TFMs' (a mean-forecast gap, distinct from the twins' ΔB, which it tracks); the extra noise cost; and a *structure part*, the TFMs' rise in $L_t(m)$ from y to y^- minus the comparator's (Table 19). With m_T and m_c the TFMs' and the comparator's mean forecasts on y (for the TFMs, losses and noise costs are averages over the four models), m_T^- and m_c^- those on y^-, and N_T and N_c their noise costs on y,

$$\underbrace{L_t(m_c) + N_c - L_t(m_T) - N_T}_{\text{advantage over the redraws}} = \underbrace{L_t(m_c^-) - L_t(m_T^-)}_{\text{linear-fit part}} + \underbrace{N_c - N_T}_{\text{extra noise cost}} + \underbrace{\big[L_t(m_T^-) - L_t(m_T)\big] - \big[L_t(m_c^-) - L_t(m_c)\big]}_{\text{structure part}}.$$

Against BART, table by table, the linear-fit part tracks the twins' ΔB (correlation 0.91) and the structure part tracks the ablation gap (correlation 0.97); against CV-tuned boosting the correlations are 0.80 and 0.65. The structure part compares forecasts averaged over the redraws, so a change in how much noise a learner fits once the structure is removed enters it only through the Monte Carlo error of the nine-redraw mean, about a ninth of that change. Its mean is close to the ablation gap against BART (0.025 and 0.026) and smaller against CV-tuned boosting (0.017 and 0.027): against BART the ablation gap reflects a change in the mean forecast, while against CV-tuned boosting part of it may reflect a change in noise fitting or the realized noise. The realized advantage exceeds the average over the redraws by 0.026 against CV-tuned boosting and 0.023 against BART: the real residual around $\hat{f}$, which the permutations scramble, still carries something the TFMs use more than

the comparators. The remainder of Figure 1 equals this difference less the gaps between the matching parts of the two splits.

Missed structure or the noise's shape? Permuting e also scrambles how the noise's size varies across rows and which rows carry its largest values. We therefore reran the redraws on y on the same 237 partitions with random sign flips instead, $y_j = P_Z y + (I - P_Z)\hat{f}(X) + (I - P_Z)(s_j \circ e)$ with independent random signs $s_j \in \{\pm 1\}^n$, which keep each row's $|e_i|$ before the projection and scramble only the signs. The realized advantage exceeds the average over these redraws by 0.023 [0.009, 0.038] against BART, against 0.023 with permutations (difference -0.0005 [-0.007, 0.006]); by 0.018 against context-tuned BART (0.017 with permutations) and by 0.021 against CV-tuned boosting (0.026; difference -0.005 [-0.013, 0.002]). Keeping each row's $|e_i|$ thus leaves the excess over BART in place and most of the excess over CV-tuned boosting: what the TFMs use more lies in the signs of the residual, consistent with structure $\hat{f}$ missed rather than with how the noise's size varies across rows. Sign flips also make skewed noise symmetric; the permutation redraws, which keep the noise's values, give a similar excess.

Table 19: Exact split of the four main TFMs' advantage averaged over the noise redraws, on the 237 partitions of Figure 1 (35 tables; 236 for context-tuned BART). Realized: the advantage on the real context. Means over tables, 95% percentile bootstrap interval for the averaged advantage, shares of it in parentheses; last column: tables on which the structure part is positive. Noise costs here use divisor 9.

Comparator	Realized	Averaged over redraws	Linear-fit part	Extra noise cost	Structure part	Structure > 0
Boosting, CV-tuned	0.072	0.046 [0.022, 0.069]	0.016 (35%)	0.013 (27%)	0.017 (38%)	29
LightGBM, Optuna-tuned	0.061	0.041 [0.018, 0.064]	0.011 (27%)	0.012 (29%)	0.018 (44%)	28
CatBoost, early-stopped	0.065	0.056 [0.031, 0.080]	0.034 (61%)	0.003 (5%)	0.019 (34%)	25
BART	0.057	0.034 [0.013, 0.056]	0.015 (43%)	-0.005 (-16%)	0.025 (73%)	24
BART, context-tuned	0.042	0.025 [0.006, 0.044]	0.009 (37%)	0.0004 (2%)	0.016 (61%)	27

Default ensembles. The four main TFMs, which run with one ensemble member per call in the main audit, are also run with eight members, their usual default, on the same twins. One- and eight-member forecasts share one code path, which for TabPFN is the regressor's point prediction rather than the mean of 257 quantile levels, so the one-member values below (a separate run with the released default settings, whose one-member values differ from the main audit's for TabPFN and TabDPT-Turbo), not those of the main audit, are the baseline for the eight-member comparison. With eight members median V falls or stays level for all four models, and mean D rises for TabPFN 2.6 and TabICLv2 and stays about level for TabPFN 3 and TabDPT-Turbo: median V and mean D go from 0.023 and 0.087 to 0.014 and 0.107 for TabPFN 2.6, from 0.008 and 0.095 to 0.006 and 0.093 for TabPFN 3, from 0.025 and 0.087 to 0.022 and 0.100 for TabICLv2, and from 0.010 and 0.098 to 0.010 and 0.096 for TabDPT-Turbo. The four main TFMs' net-discovery gap goes from 0.035 [0.010, 0.065] with one member to 0.042 [0.011, 0.079] with eight against BART, and from 0.035 [0.014, 0.055] to 0.042 [0.015, 0.068] against CV-tuned boosting. With eight members their advantage is 0.051 [0.020, 0.083] over BART and 0.063 over CV-tuned boosting; over context-tuned BART it is 0.037 with net-discovery gap 0.027 [0.002, 0.055] (0.020 [-0.001, 0.044] with one member), and over Optuna-tuned LightGBM 0.057 with 0.045 [0.019, 0.070]. The net-discovery gap over the marginalized GP is 0.029 [0.001, 0.060] with one member and 0.036 [0.0005, 0.074] with eight.

Absolute error. With absolute error normalized by the context standard deviation in place of squared error, the four main TFMs' advantage and its net-discovery part are positive against each of the five comparators checked: against BART 0.053 [0.039, 0.070] with net-discovery gap 0.038 [0.025, 0.052], against context-tuned BART 0.042 with 0.030 [0.020, 0.039], against the marginalized GP 0.050 with 0.031, against CV-tuned boosting 0.053 with 0.042 [0.031, 0.054], and against Optuna-tuned LightGBM 0.049 with 0.041.

J Ensemble and fixed-parameter controls

Core learner settings. Each learner is fitted separately to every twin-set member's context, and every tuning choice uses only that context. *Boosting, fixed*: scikit-learn's `HistGradientBoosting`

`Regressor` with 300 iterations, learning rate 0.05, at most 7 leaves, at least 5 samples per leaf, L_2 penalty 1.0 and no early stopping. *Boosting, CV-tuned*: the same regressor at learning rate 0.05, without early stopping and at the library's default L_2 penalty (zero), with iterations in $\{25, 50, 100, 200, 400\}$, leaves in $\{3, 7, 15\}$ and minimum leaf size in $\{5, 10\}$, chosen by shuffled 5-fold cross-validated MSE on the context. *CatBoost, defaults*: `CatBoostRegressor` at its library defaults, single-threaded, fixed seed. *CatBoost, early-stopped* ("context-tuned" in most tables): an 80/20 split of the context, up to 1,000 iterations at learning rate 0.05 and depth 6, stopped at the best validation iteration (patience 50), then refitted on the full context at that count. *GP, ARD*: an RBF kernel with one length scale per standardized feature, times a constant, plus white noise, all fitted by marginal likelihood (two restarts). *Boosting, CV-tuned on the least-squares residual*: the same CV-tuned booster (same grid, folds and learning rate) fitted to the residual of least squares on $[1, X]$ with features clipped at ± 4; the forecast is the least-squares fit plus the boosted residual. *Boosting, fixed, 20 bags*: the fixed booster bagged over 20 bootstrap resamples of the context. *Random forest*: 300 trees, minimum leaf size 2, half of the features considered at each split. *MLP*: standardized inputs, two hidden layers of 64 units, L_2 penalty 10^{-3}, Adam with learning rate 10^{-3}, up to 2,000 iterations and no early stopping; the *MLP, 10-member ensemble* averages ten such MLPs with different initializations on the same data. *k-NN*: standardized inputs, 10 neighbors, inverse-distance weights. *CV ridge*: ridge regression with the penalty chosen by efficient leave-one-out cross-validation over 17 values log-spaced from 10^{-4} to 10^4. *Gaussian process (marginal likelihood)*: standardized inputs, a constant times an isotropic RBF kernel (one length scale shared by all features) plus white noise, with hyperparameters maximizing the marginal likelihood with two restarts.

Controls. Controls use the main audit's $n = 120$ inputs and linear reference model (middle block of Table 12). *GP, fixed kernel, noise variance 0.1*: the marginal-likelihood GP's kernel with fixed hyperparameters, length scale $\sqrt{d}$ on standardized inputs, unit signal variance and noise variance 0.1 of the standardized target variance. *GP, fixed kernel, residual-calibrated noise* ("noise var. ν" in the tables): the same kernel with noise variance ν and signal variance $1 - \nu$, where ν is the residual fraction of the reference model's least-squares fit on the standardized context, clipped to $[0.02, 0.98]$. Since ν is a function of $T(y)$, the kernel is identical on every twin-set member and the GP's sensitivity is that of a fixed linear smoother. The bagged booster and the MLP ensemble control for ordinary variance reduction, the two fixed-kernel GPs for fixed GP hyperparameters.

K Additional partitions

Table 20 lists the sets of tables and partitions behind the results in the body. A quantity such as the advantage over BART averages a different set in each, so its value differs somewhat between figures and tables.

Table 20: Evaluation sets behind the body's results.

Set	Tables and partitions	Used for
Main partition	all 57 tables, one partition each	ranks and losses (Section 4, Appendix C), twin quantities B and D (Table 12)
Additional partitions	55 tables, up to ten partitions each	twin gaps averaged over partitions (this appendix)
Screened spare-row partitions	36 spare-row tables, 275 of 357 partitions	structure ablation (Table 2, Figure 5a,c, Table 5)
Partitions with noise redraws	35 spare-row tables, 237 partitions	accounting of Figure 1
New CTR23 tables	16 tables, screened partitions among partitions 0–9	replication (Table 2, Figure 5b)
Prior draws	feature matrices of all 57 tables, main partition	Figure 5d

Table 21 gives the twins' split of the advantage for the comparators run on every partition.

Table 21: The four main TFMs' advantage over the comparators run on every partition, split exactly as $\Delta B + \Delta D$ (Eq. (5)), with the sensitivity gap ΔV and the share $\Delta D/$advantage: means over the 55 re-partitioned tables of each table's average over up to ten partitions; and the per-table Hodges–Lehmann estimate of ΔD; 95% intervals over tables. Positive values favor the TFMs.

Comparator	advantage	ΔV	ΔB	ΔD	ΔD share	ΔD per table (HL)
BART	0.050 [0.034, 0.070]	−0.0001	0.017	0.033 [0.018, 0.052]	66%	0.017 [0.011, 0.032]
BART, context-tuned	0.035 [0.023, 0.048]	0.001	0.014	0.021 [0.011, 0.032]	60%	0.015 [0.008, 0.025]
DART (sparse BART)	0.066 [0.047, 0.085]	0.005	0.030	0.036 [0.023, 0.050]	55%	0.025 [0.017, 0.037]
GP, marginalized	0.068 [0.046, 0.092]	−0.010	0.037	0.031 [0.017, 0.047]	46%	0.020 [0.007, 0.038]
Boosting, CV-tuned	0.064 [0.048, 0.081]	0.021	0.021	0.043 [0.031, 0.057]	67%	0.037 [0.029, 0.047]

Additional partitions. Each table is re-partitioned three times with new seeds and new twins ($K = 39$, linear reference model; 55 of the 57 tables could be re-partitioned), and the audit is run for the four main TFMs, BART, DART (Appendix G), context-tuned BART, the marginalized GP and CV-tuned boosting. Table 23 reports the four main TFMs' advantage and net-discovery gap per partition and averaged per table over the three. Six further partitions are drawn the same way for each of the 55 re-partitionable tables, with the same learners (the first four draws reproduce the main and three additional partitions exactly; 52 tables had six eligible new splits). On the six new partitions alone the net-discovery gap over BART is 0.037 [0.020, 0.056] and over DART 0.041 [0.025, 0.060]. Our estimates average each table over all of its partitions (ten for 52 tables, four for the rest; Table 16). The per-table gaps over BART are heavy-tailed. The largest, 0.361, is on *CERN-Electron-Collision-Data*, whose target, an invariant mass, is a smooth function of products of the recorded energies and momenta; the next four are *analcatdata_apnea2*, *seoul_bike_sharing_demand*, *methane_emissions_rice_crop* and *black_friday*. Without the largest one, three or five tables the mean gap is 0.027, 0.022 and 0.017.

Main and additional partitions. On the 55 re-partitioned tables, the four main TFMs' normalized test MSE is 0.148 below the linear fit's on the main partition and 0.110 averaged over the additional partitions (0.114 per table over all its partitions, main included), and BART's 0.098 and 0.059 (paired difference for the TFMs 0.038 [−0.001, 0.083]). Three tables account for the difference, *Worker-sCompensation*, *analcatdata_apnea2* and *methane_emissions_rice_crop*, whose main-partition test rows favor nonlinear learners much more than their other partitions do; without them the improvements are 0.110 and 0.106 for the TFMs and 0.059 and 0.058 for BART. The main partition is thus not systematically different from the others, but its means depend more on single tables.

L Additional context sizes

New partitions are drawn from the same frozen candidate list at 60, 240, 500 and 1,000 context rows (64 audit, 120 test), with the same reference basis and twin construction; tables without enough rows drop out and are not replaced. Most of these tables are among the 57 audited ones (55 of 58 at 60 rows, 50 of 52 at 240, 44 of 47 at 500 and 40 of 43 at 1,000 rows). Table 22 lists the medians and the exact per-table split of the TFM advantage into a linear-fit gap and a net-discovery gap (Eq. (5)), as means so that the two parts add to the mean advantage; Table 24 gives bootstrap intervals for both parts and for ΔV. TFM entries are the per-table mean over the four main TFMs, then the median (or mean) over tables. The roster shrinks toward large tables as n grows, so the lower block restricts to the 38 tables present at every size; there the advantage is monotonic in n (full roster: 0.061, 0.035, 0.039, 0.020, 0.011; fixed roster: 0.074, 0.060, 0.045, 0.021, 0.013).

On the 38 tables present at every context size, the advantage over CV-tuned boosting shrinks with context, mainly through its net-discovery part: in means, the net-discovery gap falls from 0.086 at 60 rows to 0.000 at 1,000 while the linear-fit gap falls from 0.027 to 0.009 (Table 24). Medians show the same pattern (net-discovery gap 0.054 to 0.008, linear-fit gap 0.014 to 0.005). At 1,000 rows the advantage is small: its median is 0.013 [0.006, 0.020] and its mean is not distinguishable from zero.

Table 22: Additional context sizes, linear reference model, medians over tables unless marked: TFMs are the mean of the four main TFMs; boosting is CV-tuned.

n	tables	noise sensitivity V		TFM advantage	advantage $= \Delta B + \Delta D$
		TFMs	boosting	median [95% CI]	(means)
60	58	0.028	0.056	0.061 [0.027, 0.091]	$0.089 = 0.020 + 0.069$
120	57	0.016	0.033	0.035 [0.011, 0.059]	$0.060 = 0.021 + 0.039$
240	52	0.010	0.020	0.039 [0.021, 0.055]	$0.056 = 0.019 + 0.037$
500	47	0.006	0.013	0.020 [0.012, 0.035]	$0.036 = 0.014 + 0.022$
1000	43	0.004	0.008	0.011 [0.005, 0.018]	$0.008 = 0.007 + 0.001$
Fixed roster: the 38 tables present at every size					
60	38	0.028	0.056	0.074 [0.021, 0.113]	$0.113 = 0.027 + 0.086$
120	38	0.020	0.034	0.060 [0.011, 0.110]	$0.077 = 0.023 + 0.054$
240	38	0.010	0.022	0.045 [0.031, 0.074]	$0.072 = 0.021 + 0.051$
500	38	0.007	0.013	0.021 [0.007, 0.049]	$0.042 = 0.017 + 0.025$
1000	38	0.005	0.008	0.013 [0.006, 0.020]	$0.009 = 0.009 + 0.000$

Against BART, context-tuned BART and early-stopped CatBoost. Table 25 splits the four main TFMs' advantage over each (Appendix I) at every context size on the same 38 tables. Against BART, ΔV is -0.014 to 0.008 at every size (BART is the less sensitive learner at 60 rows). At 60 to 500 rows the net-discovery gap is 0.039–0.056 and the median advantage 0.026–0.038; at 1,000 rows the median advantage is 0.015. At 1,000 rows one table (OpenML *Job_Profitability*) dominates the mean advantage (0.177 [0.022, 0.460]). Its context holds one target more than 30 standard deviations below the mean; BART's leaf prior is scaled by the target range, so its real-table fit follows that point (normalized loss above 5) while the twins, whose Gaussian residual redraw contains no such point, do not. Without that table the mean advantage at 1,000 rows is 0.046 [0.017, 0.083] and the net-discovery gap 0.042; the median is unaffected. The same table dominates the booster comparison: there the tuned booster's real-table loss is 1.122 against 0.051 for the main TFMs (advantage 1.071), and excluding it moves the booster's 1,000-row mean advantage from 0.009 to -0.020 and its net-discovery gap from 0.000 to -0.030. Against context-tuned BART the median advantage is 0.013–0.047 at 60 to 500 rows and 0.021 at 1,000, ΔV is -0.004 to 0.005, and the net-discovery gap 0.033–0.050 through 500 rows; the same table dominates the 1,000-row mean, and without it the net-discovery gap is 0.047. At 60 rows tuning does not help BART: the median advantage over the tuned version (0.040) exceeds that over the defaults (0.027) and the means agree (0.052 against 0.050). Choosing among six configurations by 5-fold cross-validation on 60 rows is itself a fit to the context. The main TFMs' median advantage over early-stopped CatBoost falls from 0.060 at 60 rows to 0.013 at 1,000, as against the scikit-learn booster (0.074 to 0.013); at small n its largest component is the linear-fit gap (0.057 at 60 rows).

M Contexts of 2,000 to 10,000 rows

New partitions with 2,000, 5,000 and 10,000 context rows (64 audit and 120 test rows) are drawn from the same frozen candidate list, with the same reference basis and twin construction and $K = 19$ twins per table. Tables without enough rows drop out, leaving 35, 32 and 23 tables (32, 29 and 22 of them among the 57 main tables). The four main TFMs run as single estimators, as in the main audit. Single tables dominate the means: on OpenML 42688 every learner's normalized test loss is near 500 at 5,000 rows, and at 10,000 rows BART is the only learner below the linear fit (0.011 against 0.173; every TFM and both boosters above 0.7), which alone moves the mean advantage over BART below zero at both sizes; at 2,000 rows OpenML 44311 contributes most of the mean. Table 26 reports per-table Hodges–Lehmann estimates beside the means. Against BART, the per-table net-discovery gap is 0.020–0.024 with every interval above zero, and the TFMs have the lower loss on 77–83% of tables. The advantage over CV-tuned boosting also stays above zero; over Optuna-tuned LightGBM it falls from 0.012 [0.002, 0.023] at 2,000 rows to 0.001 [-0.008, 0.014] at 10,000, where the TFMs have the lower loss on 52% of tables. With their default eight-member ensembles at 2,000 rows, the four main TFMs' per-table advantage is 0.025 [0.009, 0.052] over BART and 0.015 [0.004, 0.026] over Optuna-tuned LightGBM, with net-discovery gaps of 0.026 and 0.013.

Table 23: The four main TFMs against the comparators on three additional partitions of 55 of the 57 tables; the other two could not be re-partitioned under the materialization checks of the main audit (advantage and net-discovery gap ΔD, 95% bootstrap intervals over tables; positive favors the TFMs).

Partition	vs BART		vs marginalized GP		vs CV-tuned boosting	
	advantage	ΔD	advantage	ΔD	advantage	ΔD
additional 1	0.033 [0.014, 0.057]	0.020 [0.003, 0.040]	0.074 [0.049, 0.103]	0.035 [0.017, 0.055]	0.058 [0.033, 0.085]	0.042 [0.020, 0.066]
additional 2	0.051 [0.024, 0.083]	0.033 [0.002, 0.068]	0.064 [0.027, 0.107]	0.017 [−0.007, 0.039]	0.073 [0.042, 0.111]	0.051 [0.027, 0.081]
additional 3	0.053 [0.033, 0.075]	0.037 [0.019, 0.056]	0.065 [0.037, 0.098]	0.025 [0.003, 0.050]	0.066 [0.039, 0.092]	0.043 [0.023, 0.064]
mean of three	0.046 [0.028, 0.066]	0.030 [0.013, 0.050]	0.068 [0.047, 0.090]	0.026 [0.013, 0.039]	0.066 [0.048, 0.086]	0.045 [0.030, 0.062]

Table 24: Fixed roster (38 tables): the TFM advantage over CV-tuned boosting split exactly into a linear-fit gap and a net-discovery gap, with the sensitivity gap ΔV; means with 95% bootstrap intervals, linear reference model; positive favors the TFMs.

n	advantage $= \Delta B + \Delta D$	linear-fit gap ΔB	net discovery gap ΔD	ΔV
60	0.113 [0.051, 0.199]	0.027 [0.007, 0.048]	0.086 [0.030, 0.172]	0.026 [0.019, 0.033]
120	0.077 [0.038, 0.118]	0.023 [−0.002, 0.049]	0.054 [0.029, 0.085]	0.020 [0.015, 0.024]
240	0.072 [0.048, 0.100]	0.021 [0.011, 0.033]	0.051 [0.028, 0.077]	0.016 [0.013, 0.020]
500	0.042 [0.023, 0.062]	0.017 [0.003, 0.039]	0.025 [0.005, 0.045]	0.011 [0.008, 0.013]
1000	0.009 [−0.078, 0.092]	0.009 [0.003, 0.015]	0.000 [−0.087, 0.085]	0.007 [0.005, 0.009]

Native pipelines at larger contexts. TabPFN 3 and TabICLv2 are also run in their native pipelines (as in Appendix I) instead of the audit's preprocessing (at most 20 features) and single estimators, on the raw features of the same partitions at each size (same seeds; tables with at least the context size plus 184 rows, without OpenML 46736, whose raw features LightGBM could not use, so the counts differ slightly from Table 26), against LightGBM tuned by Optuna as above on the same raw features (median-imputed numerics, one-hot categoricals, no feature cap), comparing test accuracy only. The two TFMs' per-table (Hodges–Lehmann) advantage is 0.015 [0.007, 0.037] at 2,000 rows (32 tables; lower loss on 84% of them), 0.017 [0.007, 0.031] at 5,000 rows (28 tables; 86%) and 0.003 [−0.006, 0.018] at 10,000 rows (23 tables; 57%). With native pipelines the TFMs keep a clear per-table edge through 5,000 rows, and no advantage is detected at 10,000 rows. This agrees with the per-table summaries of the preprocessed runs: the Hodges–Lehmann advantage over Optuna-tuned LightGBM above, and at 1,000 rows the median advantage over CV-tuned boosting of 0.013 [0.006, 0.020] (Table 22, Appendix L), with a mean, dominated by a few tables, not distinguishable from zero.

N Further results

Number of twins. Recomputing every quantity from the first 9 or 19 of the 39 stored twins, on the first three additional partitions (55 tables, 165 partitions), changes little. With 9 twins the per-table net-discovery gap over BART has correlation 0.999 (rank correlation 0.99) with the full audit's, median absolute difference 0.003; over CV-tuned boosting 0.993 (0.98), and the gap in noise sensitivity 0.962 (0.96). V, a population variance over the twins, is lower by about the factor $(K-1)/K$ (the TFMs' mean 0.016 against 0.017), which a divisor of $K-1$ removes. With 19 twins the correlations are 1.000, 0.998 and 0.981. An audit with 10 fits per table instead of 40 thus gives nearly the same gaps. The full audit of one table (the real table and 39 twins) takes a median of 2 (TabICLv2) to 27 seconds (TabPFN 3) for the four main TFMs on one GPU, and 37 seconds for BART, 124 for DART and 171 for context-tuned BART, each on one CPU core.

Table 25: Fixed roster (38 tables): the four main TFMs' advantage over BART, context-tuned BART and early-stopped CatBoost at each context size, split exactly into a linear-fit gap and a net-discovery gap, with the sensitivity gap ΔV (means with 95% bootstrap intervals, linear reference model; positive favors the TFMs), and the median advantage beside the mean. *At 1,000 rows one table (Job_Profitability) dominates the means over BART and context-tuned BART; excluding it, the advantage is 0.046 over BART and 0.052 over context-tuned BART, with net-discovery gaps 0.042 and 0.047.

Comparator	n	advantage (mean)	advantage (median)	linear-fit gap ΔB	net discovery gap ΔD	ΔV
BART (dbarts defaults)	60	0.050 [0.006, 0.096]	0.027	0.005 [−0.024, 0.030]	0.045 [0.005, 0.100]	−0.014 [−0.019, −0.008]
BART (dbarts defaults)	120	0.069 [0.035, 0.111]	0.038	0.013 [−0.007, 0.031]	0.056 [0.019, 0.106]	−0.002 [−0.005, 0.002]
BART (dbarts defaults)	240	0.061 [0.039, 0.086]	0.031	0.022 [0.012, 0.033]	0.039 [0.020, 0.061]	0.006 [0.004, 0.009]
BART (dbarts defaults)	500	0.059 [0.034, 0.089]	0.026	0.012 [0.001, 0.031]	0.046 [0.025, 0.073]	0.008 [0.006, 0.011]
BART (dbarts defaults)	1000	0.177 [0.022, 0.460]*	0.015	0.004 [0.000, 0.009]	0.173 [0.018, 0.454]*	0.008 [0.006, 0.010]
BART, context-tuned	60	0.052 [0.015, 0.091]	0.040	0.003 [−0.020, 0.024]	0.050 [0.015, 0.090]	−0.004 [−0.010, 0.003]
BART, context-tuned	120	0.051 [0.025, 0.079]	0.047	0.014 [−0.003, 0.029]	0.038 [0.008, 0.075]	0.002 [−0.002, 0.005]
BART, context-tuned	240	0.056 [0.037, 0.079]	0.036	0.021 [0.011, 0.033]	0.035 [0.019, 0.057]	0.005 [0.002, 0.008]
BART, context-tuned	500	0.046 [0.022, 0.074]	0.013	0.012 [0.000, 0.032]	0.033 [0.013, 0.058]	0.005 [0.003, 0.006]
BART, context-tuned	1000	0.159 [0.029, 0.388]*	0.021	0.005 [0.000, 0.009]	0.154 [0.024, 0.389]*	0.005 [0.003, 0.007]
CatBoost, context-tuned	60	0.094 [0.025, 0.192]	0.060	0.057 [0.020, 0.100]	0.037 [−0.014, 0.117]	0.004 [−0.001, 0.010]
CatBoost, context-tuned	120	0.086 [0.045, 0.129]	0.057	0.047 [0.022, 0.075]	0.039 [0.011, 0.073]	0.011 [0.008, 0.015]
CatBoost, context-tuned	240	0.054 [0.037, 0.072]	0.034	0.029 [0.016, 0.043]	0.026 [0.015, 0.037]	0.011 [0.009, 0.014]
CatBoost, context-tuned	500	0.045 [0.020, 0.073]	0.021	0.021 [0.006, 0.044]	0.023 [0.007, 0.039]	0.009 [0.006, 0.011]
CatBoost, context-tuned	1000	0.053 [−0.012, 0.149]	0.013	0.010 [0.003, 0.018]	0.043 [−0.022, 0.140]	0.007 [0.005, 0.009]

Table 26: Contexts of 2,000 to 10,000 rows: the four main TFMs' advantage over each comparator and its split, per-table Hodges–Lehmann estimates and means, with 95% bootstrap intervals over tables (linear reference model, $K = 19$; positive favors the TFMs). "Advantage > 0" is the share of tables on which the TFMs have the lower loss.

			per table (Hodges–Lehmann)			means		
Context	Comparator	tables	advantage	ΔD	advantage > 0	advantage	ΔD	ΔV
2,000	BART	35	0.022 [0.008, 0.048]	0.024 [0.007, 0.049]	77%	0.079 [0.018, 0.182]	0.079 [0.019, 0.182]	0.006 [0.004, 0.008]
2,000	Boosting, CV-tuned	35	0.013 [0.003, 0.030]	0.014 [0.003, 0.032]	71%	0.027 [0.007, 0.053]	0.025 [0.006, 0.049]	0.005 [0.003, 0.007]
2,000	LightGBM, Optuna-tuned	35	0.012 [0.002, 0.023]	0.011 [0.002, 0.025]	66%	0.026 [0.003, 0.056]	0.024 [0.003, 0.054]	0.004 [0.002, 0.006]
5,000	BART	32	0.021 [0.006, 0.048]	0.020 [0.002, 0.046]	78%	−0.033 [−0.193, 0.063]	−0.036 [−0.192, 0.059]	0.003 [0.002, 0.005]
5,000	Boosting, CV-tuned	32	0.015 [0.005, 0.036]	0.012 [0.004, 0.026]	75%	0.162 [0.012, 0.444]	0.172 [0.008, 0.484]	0.002 [0.001, 0.002]
5,000	LightGBM, Optuna-tuned	32	0.006 [0.000, 0.016]	0.004 [−0.003, 0.012]	66%	0.163 [−0.002, 0.480]	0.162 [−0.005, 0.484]	0.002 [0.001, 0.002]
10,000	BART	23	0.021 [0.005, 0.077]	0.020 [0.006, 0.075]	83%	−0.023 [−0.178, 0.073]	0.016 [−0.057, 0.073]	0.002 [0.001, 0.003]
10,000	Boosting, CV-tuned	23	0.017 [0.002, 0.073]	0.012 [−0.001, 0.042]	74%	0.046 [0.010, 0.090]	0.026 [0.002, 0.054]	0.001 [0.0004, 0.002]
10,000	LightGBM, Optuna-tuned	23	0.001 [−0.008, 0.014]	0.003 [−0.005, 0.013]	52%	0.010 [−0.007, 0.031]	0.011 [−0.005, 0.031]	0.001 [0.001, 0.002]