# USEFUL FEATURES CAN CROWD OUT INTERACTIONS IN TABULAR FOUNDATION MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Tabular foundation models (TFMs) can lose an interaction when other columns become informative, even though those columns remain useful at test time. We call this *crowding out*. The TFMs predict from labeled context rows with fixed pretrained weights. We add a known parity component to real regression targets and compare the original competing columns with permuted copies. The target, table width and column marginals stay fixed, separating crowding from dilution by uninformative columns. TabPFN-3, TabICLv2 and TabFM exhibit crowding; boosting and most networks trained by gradient descent benefit from the same columns. Weak additive or pairwise clues improve interaction learning without preventing crowding. Swapping labels used to construct column representations induces or releases crowding while prediction labels stay fixed. CrowdBench tests interaction retention on controlled tasks, known rules and measured systems. In selected measured systems, summaries improve RMSE while hiding component loss, or hurt prediction as context grows. These cases establish susceptibility rather than prevalence. Searching prediction residuals and refitting on the discovered columns restores planted interactions within the search dictionary.

## 1 INTRODUCTION

TabPFN, TabICL and TabFM obtain strong results with little engineering for each task (Hollmann et al., 2025; Grinsztajn et al., 2026; Qu et al., 2025; 2026; Google Research, 2026). These tabular foundation models (TFMs) make predictions from labeled context rows without updating their parameters. Providing an informative feature seems a natural way to help them. We ask whether making one part of the target easier to predict preserves what they can learn about the rest.

We find that they can lose an interaction they would otherwise capture. A model learns the interaction with uninformative competing columns, then suppresses it when those columns help predict the target. We call this *crowding out*. The columns remain predictive during training and testing; neither a distribution shift nor a failing shortcut is needed. Pure parity offers no marginal clues, but weak additive or pairwise signals improve its learnability without preventing crowding (Section 3.3, Figure 3).

Strong main effects can coexist with a component that functions of lower order cannot represent (Lengerich et al., 2020). Measured protein fluorescence contains terms of higher order (Poelwijk et al., 2019), and software runtimes depend on interacting configuration options (Siegmund et al., 2015). These mixed targets motivate CrowdBench: we ask whether making an easy component more accessible suppresses a learnable remainder, including when simpler clues are available. Appendix J discusses the supporting evidence and its limits.

We add three independent columns carrying a parity interaction to a real regression table, and add the interaction to its target. Permuting the original columns then preserves the target, width and column marginals. Across five tables with 16,000 context rows, TabPFN-3's interaction capture is $0.76$ with permuted columns but only $0.35$ with informative columns (Figure 1). Uninformative columns can also suppress an interaction; we call this *dilution*. Dilution hides crowding if it removes the interaction in both conditions. Narrow tables reveal crowding in all three TFMs, whereas boosting and networks trained by gradient descent generally benefit from the information.



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.

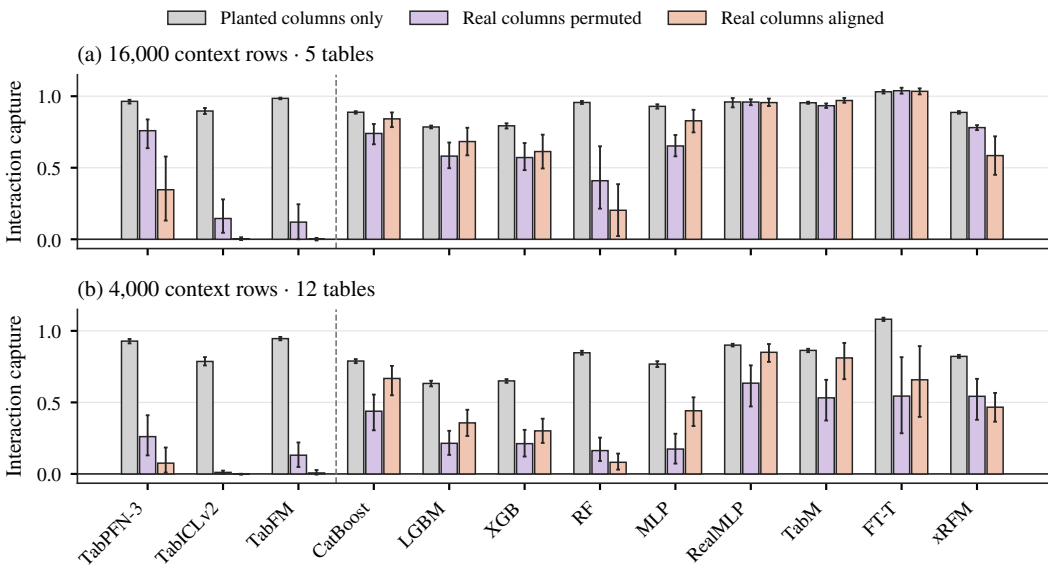


Figure 1: **Useful columns can suppress an interaction.** We measure capture of a planted parity from its three columns alone, with additional real columns permuted, and with those columns aligned. The target is identical across arms; capture is a fitted coefficient, with 1 denoting full recovery. Error bars give 95% intervals from bootstrapping tables. The drop from permuted to aligned columns measures crowding.

We next intervene on the labels used to construct column representations, keeping prediction labels fixed. CrowdBench extends the evaluation to fifteen tables with known rules, the measured CH65 antibody binding landscape, and SGEMM/KTT GPU runtimes. A summary fitted on reference rows improves CH65 RMSE while suppressing a measured component of order three. For runtimes, summaries help with short contexts but can cause substantial harm with longer ones. A separate calibrated test identifies pure third-order components in measured targets that TFMs leave unused and CatBoost captures. These examples show susceptibility in measured systems; they do not tell us how often it occurs.

The paper makes three contributions:

- **Crowding out, separated from dilution.** Matched informative/permuted comparisons distinguish crowding from dilution in three TFMs, random forests and xRFM. Varying competitor count, interaction order and simpler clues shows when the contrast is visible (Section 3).
- **A causal pathway through column labels.** Swapping these labels releases and induces crowding in all three TFMs, whereas perturbations of equal variance do not. Transfers at the TabPFN-3 readout implicate both attention routing and representations. Learners that allocate capacity according to marginal label association exhibit related behavior (Section 4).
- **CrowdBench and a repair by residual search.** CrowdBench evaluates interaction retention on controlled and measured tasks, and on tasks with known rules. Searching cross-fitted residuals and refitting on the discovered columns recovers suppressed interactions within the search dictionary. With a significance guard, we observe no RMSE harm above 1% on 36 native and 17 discrete tables (Sections 5 and 6). We provide CrowdBench code in the supplementary material.

## 2 Related work

**Tabular foundation models.** TabPFN and TabICL amortize inference over synthetic priors (Müller et al., 2022; Hollmann et al., 2023; 2025; Qu et al., 2025), as do their successors and TabFM (Grinsztajn et al., 2026; Qu et al., 2026; Google Research, 2026). Context examples adapt predictions to a new table without changing pretrained weights. Recent studies examine the representations and computations behind this adaptation (McCarter, 2024; Biloš et al., 2026; Gupta et al., 2026; Hu

& Ghelichi, 2026). We ask how context labels govern competing signals, intervening separately on column construction, attention routing and representations.

**What tabular benchmarks measure.** TabArena compares predictive performance across a broad collection of tables (Erickson et al., 2025). TabReD finds that model rankings can change with feature richness (Rubachev et al., 2025), and irrelevant columns are known to hurt neural tabular models (Grinsztajn et al., 2022). We instead consider informative columns. CrowdBench varies the information available to a learner while matching the target and table width. Its component measurements expose losses that aggregate accuracy can obscure, much as structural fidelity does for generated tables (Jiang et al., 2026).

**Feature competition.** Simplicity bias, gradient starvation and shortcut learning describe ways neural networks favor easy signals (Shah et al., 2020; Pezeshki et al., 2021; Hermann & Lampinen, 2020; Geirhos et al., 2020). Retraining the last layer can improve predictions using information already present in a model's representations (Kirichenko et al., 2023). This distinguishes what representations contain from what predictions use. In our setting the TFM weights are frozen, competition arises during inference, and the competing signals remain predictive at test time. Forests can likewise miss pure interactions when marginal effects dominate (Wright et al., 2016; Blum et al., 2024). AGOP feature learning and the residual fits used by boosting offer contrasting accounts of how capacity is allocated (Radhakrishnan et al., 2024; Beaglehole et al., 2025; Friedman, 2001).

**Interaction discovery and residual correction.** Because a parity has no Fourier coefficients of lower order, marginal evidence is insufficient for finding its support (O'Donnell, 2014; Barak et al., 2022; Abbe et al., 2023). Epistasis benchmarks use this property (Urbanowicz et al., 2012). Methods for interaction attribution characterize the fitted predictor (Tsang et al., 2018; Kang et al., 2025; Jia et al., 2026); residual tests instead use the observed labels to identify its omissions (Shah & Bühlmann, 2018; Kim et al., 2019). We take the latter approach, combining randomized pair search and sparse Fourier learning (Thanei et al., 2018; Linial et al., 1993) with restriction to the discovered support to remove dilution.

## 3 Understanding interaction crowding

### 3.1 Problem setting

Consider a target $y = g(x_S)+c(x_C)+\varepsilon$, where $g$ is an interaction on columns $S$ and $c$ is a competing signal. To measure how much of the interaction a predictor uses, we regress its predictions on $(1, g, c)$ by least squares on held-out rows. We call the coefficient of $g$ its *capture* $\kappa$. Capture is 1 for the Bayes predictor and 0 for a predictor independent of the interaction. It is a coefficient, not a probability, and can lie outside the unit interval.

**A matched comparison.** For a real table, we first standardize the target with statistics from the training rows. We draw three independent Gaussian columns $z$ and define

$$g(z) = 0.6\,\mathrm{sign}(z_1 z_2 z_3), \qquad y = g(z) + y_{\mathrm{real}}.$$

We predict the same $y$ in three arms: from $z$ alone (NONE), from $z$ with independently permuted real columns (PERM), and from $z$ with the original real columns (ALIGNED). Crowding is the paired difference $\kappa_{\mathrm{PERM}} - \kappa_{\mathrm{ALIGNED}}$, which holds width and column marginals fixed. Dilution is the loss of capture from NONE to PERM.

### 3.2 Informative columns suppress learnable interactions

Our benchmark has 12 regression tables with 5–136 numeric columns. Each uses 4,000 training rows; the five large tables also use 16,000. Each condition has three seeds and 2,000 test rows. We average seeds within tables, then bootstrap whole tables for 95% intervals.

TabPFN-3, TabICLv2 and TabFM are released checkpoints from independently developed families pretrained on synthetic priors. Their column or cell representations use context labels, which we can intervene on (Section 4); TabPFN-2.5 behaves similarly to TabPFN-3 (Appendix A). Nine trained

Table 1: **Crowding by model, target and context size.** Mean capture in the permuted arm minus that in the aligned arm, with 95% intervals from bootstrapping tables. Positive values (red, bold) indicate suppression by informative columns. We include every cell, including those at the dilution floor.

| | Parity | | Smooth product |
|---|---|---|---|
| Model | 4,000 rows | 16,000 rows | 4,000 rows |
| TabPFN-3 | **+0.19** [+0.07, +0.31] | **+0.41** [+0.23, +0.55] | **+0.01** [-0.01, +0.03] |
| TabICLv2 | **+0.01** [+0.00, +0.03] | **+0.14** [+0.04, +0.29] | **+0.10** [+0.06, +0.16] |
| TabFM | **+0.12** [+0.05, +0.21] | **+0.12** [0.00, +0.25] | **+0.02** [-0.01, +0.06] |
| CatBoost | -0.23 [-0.32, -0.16] | -0.10 [-0.15, -0.06] | -0.21 [-0.26, -0.16] |
| LightGBM | -0.14 [-0.19, -0.10] | -0.10 [-0.16, -0.05] | -0.04 [-0.09, +0.01] |
| XGBoost | -0.09 [-0.13, -0.04] | -0.04 [-0.09, +0.01] | -0.06 [-0.10, -0.02] |
| RandomForest | **+0.08** [+0.03, +0.13] | **+0.21** [+0.13, +0.30] | **+0.10** [+0.06, +0.14] |
| MLP | -0.27 [-0.35, -0.18] | -0.18 [-0.23, -0.11] | -0.16 [-0.29, -0.07] |
| RealMLP | -0.22 [-0.35, -0.11] | +0.00 [-0.02, +0.03] | -0.19 [-0.34, -0.09] |
| TabM | -0.28 [-0.41, -0.18] | -0.04 [-0.05, -0.01] | -0.11 [-0.22, -0.03] |
| FT-Transformer | -0.11 [-0.21, -0.02] | +0.00 [-0.03, +0.02] | -0.17 [-0.27, -0.08] |
| xRFM | **+0.08** [-0.10, +0.24] | **+0.20** [+0.08, +0.31] | **+0.05** [-0.13, +0.17] |

baselines cover different allocation mechanisms. CatBoost, LightGBM and XGBoost fit successive trees to residuals; a random forest selects splits by immediate gain. MLP, RealMLP, TabM and FT-Transformer fit columns jointly by gradient descent, with FT-Transformer providing attention without pretraining. xRFM learns a metric that reweights columns. We use released TFM defaults and the baselines' documented default or tuned configurations (Appendix A).

**All three TFMs lose capture.** Given only the planted columns, TFMs recover almost the whole interaction (0.90–0.99 at 16,000 rows). Across both sample sizes at full width, permuted minus aligned capture averages $0.25$ $[0.13, 0.36]$ for TabPFN-3, $0.12$ $[0.04, 0.21]$ for TabFM and $0.05$ $[0.01, 0.10]$ for TabICLv2 (Table 1; Figure 6 reports each table). For TabICLv2 and TabFM, mean capture with permuted columns is already at most $0.15$, leaving little room for a further decrease. Thus, a small contrast at full width can arise from two poor fits.

Retaining only two or four real columns exposes the hidden crowding. Across all 24 table–width cells, mean crowding is $0.30$ $[0.19, 0.42]$, $0.34$ $[0.25, 0.43]$ and $0.30$ $[0.22, 0.39]$ for TabPFN-3, TabICLv2 and TabFM, against $-0.06$ for CatBoost (Appendix B). More context also leaves a gap: quadrupling context on the five common tables raises TabPFN-3 capture from $0.40$ to $0.76$ with permuted columns, but only from $0.15$ to $0.35$ with aligned columns.

**The effect depends on the learner.** Boosting and networks trained by gradient descent generally use informative columns to improve capture. Among 88 cells where these baselines learn the interaction, selected using a separate seed, crowding averages $-0.14$ and never exceeds $0.11$. Random forests and xRFM, however, lose $0.21$ and $0.20$ at 16,000 rows. Both can favor columns with immediate label association, motivating the allocation interventions in Section 4. A parity column alone offers no immediate impurity reduction for a forest split. In xRFM, gradients of the fitted predictor determine the metric, and weight moves from the parity support to competing columns (Appendix F). Boosting fits residuals, allowing later trees to see the interaction after accounting for competitors. We also observe TFM crowding on 22 additional tables and in binary classification when the interaction is learnable (Appendix B).

### 3.3 Which interactions are crowded?

**Few informative columns suffice, and higher orders are more vulnerable.** One informative column costs TabICLv2 $0.40$ in capture and TabFM $0.27$ on the five large tables at 16,000 rows; two cost $0.52$ and $0.53$ (Figure 2a). TabPFN-3 crowds more as width increases, while dilution limits the other TFMs at full width. Joint row permutation preserves dependence among competitors and reproduces the contrast from independent permutations (Appendix B).

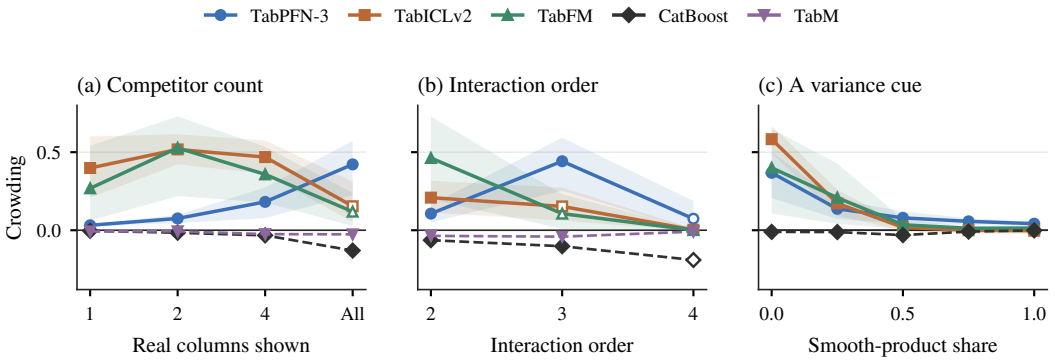


Figure 2: **Crowding is strongest when individual columns give few clues.** (a) Varying the number of real columns with a parity on three inputs. (b) Varying parity order with all real columns. Hollow markers denote capture below 0.2 in the permuted arm, where dilution limits the contrast. (c) Blending parity with a smooth product introduces variance cues. Five tables, 16,000 rows; 95% intervals bootstrap tables. In (c), TabPFN-3 sees all real columns and the other models two.

Interaction order also matters. TabPFN-3 retains most of a pairwise parity ($0.95 \to 0.85$), but much less of a parity on three inputs ($0.77 \to 0.33$). TabFM and TabICLv2 crowd pairs ($0.99 \to 0.53$ and $0.76 \to 0.56$), while dilution removes their parities on three inputs. At order four all TFMs reach the dilution floor (Figure 2b). Accuracy reflects the omission: with informative columns and a parity on three inputs, TabM beats the best TFM in all 15 table–seed cells on fresh seeds, with 19% lower mean RMSE.

**Lower-order clues need not protect the interaction.** A parity on independent symmetric inputs is independent of every proper subset of those inputs (Appendix M.1). Blending it with a smooth product supplies variance cues and rapidly removes crowding (Figure 2c). Rules dominated by main effects or pairs behave similarly (Appendix B), but these comparisons change the function to be learned. We therefore keep the interaction *fixed* and add clues of lower order. We use $g = 0.6\,\mathrm{sign}(z_1 z_2 z_3)$ and $y = g + c + q$, with $c$ the standardized real outcome and four real competing columns either aligned or jointly permuted. The additive clue

$$q = 0.6\sqrt{\rho/(1-\rho)}\,(z_1 + z_2 + z_3)/\sqrt{3}$$

accounts for population fraction $\rho \in \{0, .01, .03, .10, .25\}$ of the variance of $g + q$. For a pairwise clue, we use $(z_1 z_2 + z_1 z_3 + z_2 z_3)/\sqrt{3}$ at $\rho = .10$. We measure capture controlling for $(1, g, c, q)$, and construct hidden clues by replacing the $z$ in $q$ with independent, unobserved inputs of the same distribution. The confirmation comprises 1,980 fits across three source tables, two context sizes, three fresh seeds and five learners. We retain them all (Appendix C).

**Clues improve learning before they remove suppression.** With 16,000 rows, a 3% additive clue raises TabICLv2 capture in the permuted arm from 0.42 to 0.84. Capture with aligned columns is still 0.38, leaving a paired loss of 0.46 $[0.35, 0.57]$. Even a 10% pairwise clue leaves 0.22 $[0.08, 0.35]$ of loss. At 4,000 rows the 3% clue raises TabFM reference capture from 0.14 to 0.81, exposing a loss of 0.45 $[0.22, 0.69]$ compared with 0.12 without a clue. TabPFN-3 improves more steadily: its loss drops from 0.14 $[0.09, 0.19]$ with a 1% clue to 0.07 $[0.04, 0.09]$ at 25% (Figure 3). Clues can therefore reveal crowding by lifting the reference fit out of dilution before they remove suppression. Their effect varies with the learner; there is no sharp division between targets with no clues and ordinary targets.

## 4 A CAUSAL PATHWAY THROUGH COLUMN LABELS

Context labels enter both representation construction and prediction. On controlled tasks, knowing the interaction and its competitor lets us change the labels used to build column or cell representations while keeping prediction labels fixed.

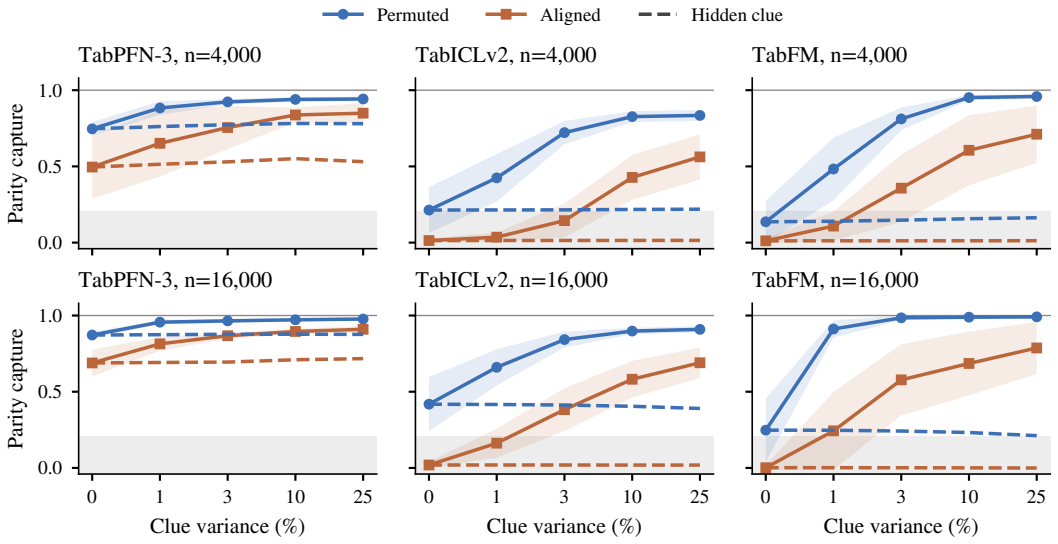


Figure 3: **Clues can improve learnability without eliminating crowding.** The parity is fixed; an additive clue supplies fraction $\rho$ of their combined variance. Solid curves show visible clues with permuted or aligned competitors. Bands give conditional 95% intervals over nine table/seed units. Dashed curves hide clues drawn from the same distribution. Gray regions indicate low reference capture. Appendix C includes CatBoost, TabM and pairwise clues.

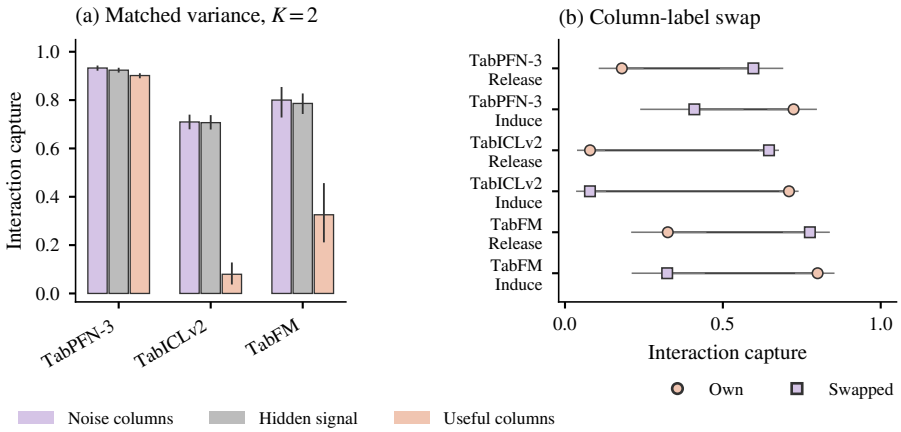


Figure 4: **Changing the column labels reverses crowding.** (a) Capture with noise columns, informative columns, and a hidden nuisance of equal variance. (b) Removing the competitor from column labels releases suppression; the reverse swap induces it. Labels for prediction remain fixed. Swaps use real competitors for TabPFN-3 and two Gaussian competitors for the others. Intervals bootstrap synthetic tasks or real table–seed units.

**Target variance does not explain the suppression.** We use $y = g + 0.3 \sum_{j \le K} x_j^{\text{comp}} + \varepsilon$, Gaussian competitors, 16,000 context rows and ten seeds. At $K = 2$, TabICLv2 captures $0.71$ with noise columns and $0.08$ with informative columns. Replacing their signal with an unobserved nuisance of equal variance restores capture to $0.71$. TabFM gives $0.80$, $0.33$ and $0.79$ (Figure 4a). These linear competitors barely affect TabPFN-3, but real columns reduce capture from $0.72$ under joint row permutation to $0.18$ under alignment. Added target noise therefore cannot explain the loss; access to competing information matters.

**Label swaps release and induce the effect.** We substitute column labels, fixing their mean and variance, inputs and prediction labels. Disabling TabPFN-3's target power transform in both arms

Table 2: **CrowdBench measured track: aligned versus permuted reference summaries.** CH65 G189E: capture loss for the selected component of order three and RMSE change (512 context rows, 24 seeds; selection uses reference rows). Runtimes: RMSE change with long contexts (2,048 rows over eight SGEMM blocks; 1,024 over six KTT contexts). At 128 rows every learner improves in all five settings, by 19–45%. FPS: grouped summary, 512 rows, eight contexts; selected by screening 31 families. $^{*}$Adjusted evidence for capture loss (CH65) or RMSE harm above 3% (TFMs, prespecified runtime families). $^{\dagger}$The adjusted $t$ interval excludes zero, but the adjusted bootstrap interval narrowly includes it. $^{\ddagger}$TabM barely fits native FPS at 512 rows (RMSE 17 times TabFM's). CatBoost and TabM represent boosting and gradient training as in Section 3. CH65 reports the G189E full summary from a grid of three assays and four summaries; FPS is one of four confirmed families. Appendices H–I report all cells and intervals.

| Task | Endpoint | TabPFN-3 | TabICLv2 | TabFM | CatBoost | TabM |
|---|---|---|---|---|---|---|
| CH65 G189E | capture loss | $0.18^{*}$ | $0.15^{*}$ | $0.16^{\dagger}$ | $0.13^{*}$ | $0.33^{*}$ |
| CH65 G189E | $\Delta$RMSE (%) | $-14.9$ | $-12.2$ | $-15.3$ | $-27.0$ | $-10.8$ |
| SGEMM | $\Delta$RMSE (%) | $+4.5$ | $+6.1$ | $-4.4$ | $-31.1$ | $+12.7$ |
| KTT GEMM / 1070 | $\Delta$RMSE (%) | $+2.0$ | $+6.1$ | $+4.9$ | $+0.3$ | $+6.1$ |
| KTT GEMM / 2080 | $\Delta$RMSE (%) | $+31.2^{*}$ | $+23.8$ | $+23.8^{*}$ | $+4.6$ | $+38.2$ |
| KTT $N$-body / 1070 | $\Delta$RMSE (%) | $+5.8$ | $-3.4$ | $-1.7$ | $-14.3$ | $-3.2$ |
| KTT $N$-body / 2080 | $\Delta$RMSE (%) | $-2.9$ | $+19.5^{*}$ | $+10.9$ | $-17.0$ | $+1.6$ |
| FPS | $\Delta$RMSE (%) | $+22.7$ | $+5.3$ | $-4.2$ | $-55.6$ | $-76.1^{\ddagger}$ |

makes this exact (Appendix D). Removing the competitor from column labels raises capture from $0.08$ to $0.65$ for TabICLv2, $0.33$ to $0.78$ for TabFM, and $0.18$ to $0.60$ for TabPFN-3. The reverse substitution reduces it from $0.71$ to $0.08$, $0.80$ to $0.32$, and $0.72$ to $0.41$, respectively. Perturbations of equal variance, independent of observed columns, change capture by at most $0.05$ (Table 3). Label content thus controls suppression, with similar results for a nonmonotone interaction (Appendix D).

We further decompose TabPFN-3's final feature aggregation block, transferring attention probabilities $A$, projected values $V$ and query skip $R$ from a donor with swapped labels. This analysis covers TabPFN-3; label swaps cover all three TFMs. Across three tables and four fresh seeds, routing alone restores $0.037$ of the full $0.416$ parity gain. With donor values and skip present, it adds $0.322$. Matched random label directions have much smaller effects (Appendix E).

**Explicit allocation produces the same signatures.** We test the account of Section 3.2 in learners where allocation can be disabled. Turning off xRFM's AGOP weighting changes crowding from $0.20$ $[0.07, 0.31]$ to $-0.08$ $[-0.14, -0.04]$; both versions learn the interaction with permuted columns. A residual forest refit reduces crowding from $0.23$ to $0.10$, and swapping labels used to grow splits releases and induces it (Appendix F). A model favoring individually predictive columns reproduces these signatures (Appendix M.2). The TFM swaps identify a causal pathway through column labels, leaving the full computation unresolved: TabICLv2 tokens spanning three columns can contain the entire interaction, and stronger competitors need not increase crowding.

## 5 CrowdBench: interaction retention under competing information

CrowdBench brings these tests together in three tracks. In the *controlled* track, planted parities and variants in width, order, rule and clues give exactly identifiable capture (Section 3). The *known-rule* track has fifteen public classification tables, six from GAMETES epistasis benchmarks (Appendix K.2). For the *measured* track, we fit a summary of lower order on separate reference rows and add it to the native predictors. We compare aligned and permuted summaries, keeping targets, native inputs, width and summary marginals fixed. Permutations are made within context and query rows; fits using native inputs alone provide another reference. We measure RMSE and component capture where it is identifiable. Users can apply the probe to their own table: it plants a parity on new columns and fits twice, with original columns aligned and permuted. The supplementary material contains this probe, code for known rules, protocols and records of each fit.

**Accuracy can hide component loss.** CH65 records antibody binding for every combination of 16 mutations against three antigens (Phillips et al., 2023). On reference rows we select the largest coefficient of order three and fit a ridge summary of main effects and pairs. After removing main effects and pairs, capture is the held-out coefficient of the selected product in predictions, divided by its coefficient in the measured outcome. For G189E, the summary reduces capture in all 24 seeds for TabPFN-3 and TabICLv2 and in 23 for TabFM. The loss is $0.15$–$0.18$, despite a 12–15% improvement in RMSE (Table 2); TabFM passes only the adjusted $t$ interval. CatBoost and TabM exhibit the same tradeoff. On MA90, a summary restricted to columns outside the scored support reduces TabICLv2 capture from $0.53$ to $0.43$ and increases RMSE by $3.4\%$. Both survive adjustment for multiple comparisons. Results vary by assay and context: TabFM capture improves on MA90, while both arms approach unit capture on G189E and SI06 at 2,048 rows. Appendix H describes the scope of this selected case study.

**Useful summaries can hurt at longer contexts.** SGEMM contains GPU runtimes for 241,600 kernel configurations (Paredes & Ballester-Ripoll, 2017). We use eight previously unused disjoint blocks and fix endpoints in advance. A ridge summary of main effects and pairs helps all five learners at 128 context rows. At 2,048 rows, however, it increases TabPFN-3 and TabICLv2 RMSE over permuted summaries by $4.5\%$ and $6.1\%$. Against the same comparator, an exact orthogonal decomposition of squared error on the query design shows increases of $11.4\%$ and $17.4\%$ outside the span of main effects and pairs (Holm-adjusted $p < .001$). This error includes measurement noise as well as any suppressed component of higher order. The RMSE increases fail the adjusted test for harm above $3\%$; TabFM and CatBoost improve, while TabM worsens. We also evaluate the independent Kernel Tuning Toolkit (Hozzová et al., 2021). A rule based only on inputs, fixed before prediction, admits GEMM and $N$-body kernels on two GPUs. All 20 learner/cell combinations improve at 128 rows. At 1,024 rows each TFM has adjusted harm in one of four cells. For RTX 2080 GEMM, their RMSE rises 24–31% over permuted summaries and 35–39% over native inputs. A grouped summary also raises TabPFN-3 RMSE by $22.7\%$ on video game FPS, selected by screening 31 families (Peeters et al., 2021). Trained baselines can suffer these harms too, and the controlled pathway need not explain them. Replacing SGEMM column labels with the target minus the summary worsens aligned prediction for TabPFN-3 and TabICLv2 (Appendix I). Here the residual subtracts an estimated summary of a target with dependent inputs; the controlled donors subtract a known competitor.

**Measured landscapes contain unused pure interactions.** We also test measured targets directly, without planting components or adding summaries. On tables with only discrete inputs, exact removal of main effects and pairs supports a test on separate samples for pure third-order structure in the model residual. We calibrate null rejection rates using each table's covariates. Of 11 calibrated tables, each TFM leaves such a component on 4–5 and CatBoost on 2. On CH65 G189E, CR9114-H3 and an HDV ribozyme landscape, TFM capture of the same scored component is only $-0.07$ to $0.41$, against $0.78$–$0.93$ for CatBoost (Appendix J). The TFMs fully capture the signal of lower order, except for an incorrectly scaled TabPFN-3 base on CR9114-H3. These omitted components are small, with median $0.1$–$0.3\%$ of residual variance. The test establishes their omission without identifying its cause.

## 6 Recovering suppressed interactions

Predictions alone cannot distinguish $Y = f(X) + \varepsilon$ from $Y = f(X) + ah(X) + \varepsilon$. Observed labels expose the difference through $R = Y - f(X)$: for a candidate $h$, adding $wh$ can reduce squared risk by at most $\mathbb{E}[Rh]^2/\mathbb{E}[h^2]$ (Appendix M). We search residuals from two-fold cross-fitting for products of three columns transformed by median signs or centered ranks. Balanced random contractions correct repeated indices exactly, scoring candidate triples in $O(Lnp^2 + Bnp)$ time instead of $O(np^3)$ (Appendix K). We refit the same TFM to these residuals on the discovered columns and add its output to the base prediction. As in the label swaps, residualization removes competing signal from the labels; using only the discovered columns removes dilution.

**Recovery within the dictionary and validation-guarded correction.** The search finds the exact support in 150 of 153 aligned parity fits. For diluted TFMs, residual refits give capture $0.00$–$0.19$ on all columns and $0.84$–$1.07$ on recovered columns, within $0.03$ of the oracle (Figure 5).

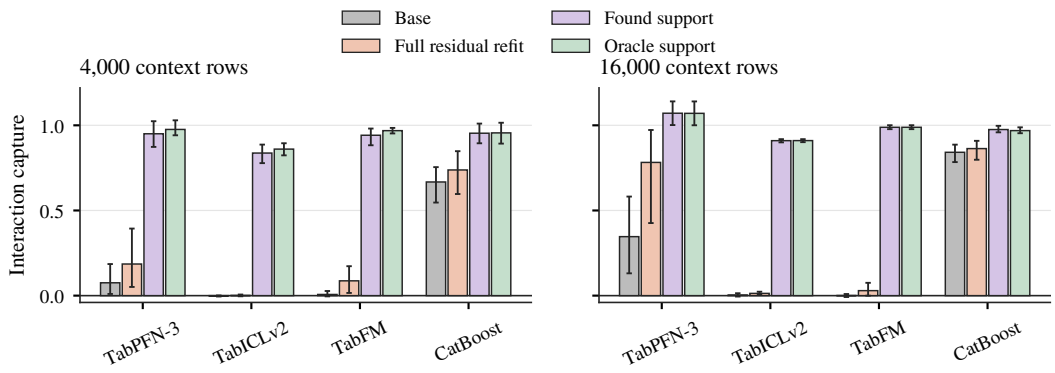


Figure 5: **Support restriction restores the interaction.** Capture on every table in the aligned parity benchmark, with intervals from bootstrapping tables. Refitting on the found support nearly matches the oracle. Refitting residuals on all columns recovers less where dilution has removed the interaction.

A separate study with independent samples and 512 columns finds RMSE improvements over refitting all columns of $10.5\%$ for TabICLv2 and $12.5\%$ for TabPFN-3. Matched random supports offer almost no gain. On the GAMETES table with three-way epistasis, TFMs have log loss $0.62$–$0.69$ (chance $0.69$). Search finds the epistatic SNPs in every split, and a correction gated by validation lowers log loss 12–20%. On the unmodified robot arm simulator `puma32H`, using a discovered support raises capture of a component of higher order from $0.08$–$0.09$ to $0.85$–$1.00$ and lowers RMSE 22–25%. Permuting other columns changes capture by less than $0.001$, identifying dilution as the source of this omission (Appendix L). Recovery still depends on the dictionary: search finds $0/10$ supports of an all-even interaction, and ridge on a single discovered parity product can match the TFM correction (Appendix K.1).

We deploy a correction only if its cross-fitted training predictions pass a paired significance test and improve MSE by at least 1%. The guard preserves planted and `puma32H` gains, with no test RMSE harm above 1% on the tested 36 native and 17 discrete tables (Appendix K.3). Gains on native tables mostly come from generic residual stacking.

## 7 Discussion

The matched tests distinguish crowding from dilution. CrowdBench measures interaction retention alongside aggregate accuracy, which can hide component loss. Engineered features should be evaluated at the deployed context size: a helpful summary can become harmful as context grows. Trained baselines share some of the measured harms.

**Pretraining and learned weights.** Further TabICLv2 training on the unchanged public generator, without parity tasks, substantially reduces classification suppression (Appendix G), but costs reference capture and ordinary-table log loss. For this model and task type, weights and optimization matter: suppression is not inevitable under the architecture. The generator's causal role remains unresolved.

Changing the synthetic prior is one candidate intervention. Generators could pair pure interactions of higher order with strong additive competitors, varying variance share, width and simpler clues. Evaluation should combine matched aligned/permuted tests with accuracy on ordinary tables.

**Limitations.** Controlled components are designed, and the TFM findings concern three recent families. Measured evidence covers one antibody system, SGEMM, two GPU kernels on two devices, and an FPS table selected by screening. These cases establish susceptibility rather than prevalence; their harms need not share the controlled pathway. We identify a pathway through column labels, leaving the full computation unresolved. Readout results concern only TabPFN-3. Recovery depends on the search dictionary, and native-table gains are mostly generic.

## REPRODUCIBILITY STATEMENT

Dataset preparation, model versions and configurations are listed in Appendix A. The appendix for each experiment specifies its protocol, seeds, sample sizes and uncertainty procedure. We provide an anonymized CrowdBench implementation, experiment scripts and saved predictions for the primary benchmark in the supplementary material. The bundle contains the probe requiring two fits, residual search, the known-rule track, protocols, records of each fit, analysis outputs and file hashes. Appendix N explains how to reproduce scoring from the saved predictions. We will make CrowdBench public after anonymous review.

## AI USE DISCLOSURE

We used large language models (LLMs) for writing and grammar checks, experimental design, implementation, local execution of experiments, analysis and interpretation, artifact repackaging and internal review. We verified the work through code and analysis review, recomputed primary benchmark metrics from saved predictions using the supplied scoring scripts, and checked claims and citations against experiment records and original sources. The authors remain responsible for the content and its final verification.

## REFERENCES

Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: Leap complexity and saddle-to-saddle dynamics. *arXiv preprint arXiv:2302.11055*, 2023. URL https://arxiv.org/abs/2302.11055.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2207.08799.

Daniel Beaglehole, David Holzmüller, Adityanarayanan Radhakrishnan, and Mikhail Belkin. xRFM: Accurate, scalable, and interpretable feature learning models for tabular data. *arXiv preprint arXiv:2508.10053*, 2025.

Devin P. Bendixsen, James Collet, Bjørn Østman, and Eric J. Hayden. Genotype network intersections promote evolutionary innovation. *PLoS Biology*, 17(5):e3000300, 2019.

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

Ricardo Blum, Munir Hiabu, Enno Mammen, and Joseph Theo Meyer. Pure interaction effects unseen by random forests. *arXiv preprint arXiv:2406.15500*, 2024.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Dheeru Dua and Casey Graff. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2019.

Nick Erickson et al. TabArena: A living benchmark for machine learning on tabular data. In *NeurIPS Datasets and Benchmarks Track*, 2025.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Google Research. Introducing TabFM: A zero-shot foundation model for tabular data. Google Research blog and model release, 2026. Model release tabfm-1.0.0.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *NeurIPS Datasets and Benchmarks Track*, 2022.

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

Atharva Gupta, Dhruv Kumar, Murari Mandal, and Saurabh Deshpande. Where computation lives inside TabPFN: Causal localisation of attention head function. *arXiv preprint arXiv:2606.12917*, 2026. URL https://arxiv.org/abs/2606.12917.

Katherine L. Hermann and Andrew K. Lampinen. What shapes feature representations? Exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems*, 2020.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2207.01848.

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*, 637:319–326, 2025.

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 (NeurIPS)*, 2024.

Jana Hozzová, Jiří Filipovič, Amin Nezarat, Jaroslav Ol'ha, and Filip Petrovič. Searching CUDA code autotuning spaces with hardware performance counters: Data from benchmarks running on various GPU architectures. *Data in Brief*, 39:107631, 2021. doi: 10.1016/j.dib.2021.107631.

James Hu and Mahdi Ghelichi. Noise immunity in in-context tabular learning: An empirical robustness analysis of TabPFN's attention mechanisms. *arXiv preprint arXiv:2604.04868*, 2026.

Jingyun Jia, Chandan Singh, Rich Caruana, and Ben Lengerich. Selecting feature interactions for generalized additive models by distilling foundation models. *arXiv preprint arXiv:2604.13332*, 2026.

Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. TabStruct: Measuring structural fidelity of tabular data. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.11950.

Justin Singh Kang, Landon Butler, Abhineet Agarwal, Yigit Efe Erginbas, Ramtin Pedarsani, Bin Yu, and Kannan Ramchandran. SPEX: Scaling feature interaction explanations for LLMs. In *International Conference on Machine Learning*, pp. 28878–28903, 2025. URL https://proceedings.mlr.press/v267/kang25a.html.

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 (NeurIPS)*, 2017.

Michael P. Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2019. URL https://arxiv.org/abs/1805.12317.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Zb6c8A-Fghk.

Benjamin Lengerich, Sarah Tan, Chun-Hao Chang, Giles Hooker, and Rich Caruana. Purifying interaction effects with the functional ANOVA: An efficient algorithm for recovering identifiable additive models. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2402–2412. PMLR, 2020. URL https://proceedings.mlr.press/v108/lengerich20a.html.

Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

Calvin McCarter. What exactly has TabPFN learned to do? ICLR Blogposts, 2024. URL https://iclr-blogposts.github.io/2024/blog/what-exactly-has-tabpfn-learned-to-do/.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2112.10510.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL https://arxiv.org/abs/2105.10386.

Enrique Paredes and Rafael Ballester-Ripoll. SGEMM GPU kernel performance. UCI Machine Learning Repository, 2017. URL https://doi.org/10.24432/C5MK70.

Yeonwoo Park, Brian P. H. Metzger, and Joseph W. Thornton. The simplicity of protein sequence-function relationships. *Nature Communications*, 15:7953, 2024. doi: 10.1038/s41467-024-51895-5.

Sven Peeters, Vitalik Melnikov, and Eyke Hüllermeier. Performance prediction for hardware-software configurations: A case study for video games. In *Advances in Intelligent Data Analysis XIX*, pp. 222–234. Springer, 2021. doi: 10.1007/978-3-030-74251-5_18.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, 2021.

Angela M. Phillips, Katherine R. Lawrence, Alief Moulana, Thomas Dupic, Jeffrey Chang, Milo S. Johnson, Ivana Cvijović, Thierry Mora, Aleksandra M. Walczak, and Michael M. Desai. Binding affinity landscapes constrain the evolution of broadly neutralizing anti-influenza antibodies. *eLife*, 10:e71393, 2021.

Angela M. Phillips, Daniel P. Maurer, Caelan Brooks, Thomas Dupic, Aaron G. Schmidt, and Michael M. Desai. Hierarchical sequence-affinity landscapes shape the evolution of breadth in an anti-influenza receptor binding site antibody. *eLife*, 12:e83628, 2023. doi: 10.7554/eLife.83628.

Frank J. Poelwijk, Michael Socolich, and Rama Ranganathan. Learning the pattern of epistasis linking genotype and phenotype in a protein. *Nature Communications*, 10:4213, 2019. doi: 10.1038/s41467-019-12130-8.

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 (NeurIPS)*, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.05564.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024.

Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, and Artem Babenko. TabReD: Analyzing pitfalls and filling the gaps in tabular deep learning benchmarks. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/571799482291411607c54984153190b0-Abstract-Conference.html.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, 2020.

Rajen D. Shah and Peter Bühlmann. Goodness-of-fit tests for high-dimensional linear models. *Journal of the Royal Statistical Society: Series B*, 80(1):113–135, 2018. URL https://arxiv.org/abs/1511.03334.

Norbert Siegmund, Alexander Grebhahn, Sven Apel, and Christian Kästner. Performance-influence models for highly configurable systems. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 284–294. ACM, 2015. doi: 10.1145/2786805.2786845.

Gian-Andrea Thanei, Nicolai Meinshausen, and Rajen D. Shah. The xyz algorithm for fast interaction search in high-dimensional data. *Journal of Machine Learning Research*, 19(37):1–42, 2018. URL https://jmlr.org/papers/v19/16-515.html.

Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1705.04977.

Ryan J. Urbanowicz, Jeff Kiralis, Nicholas A. Sinnott-Armstrong, Tamra Heberling, Jonathan M. Fisher, and Jason H. Moore. GAMETES: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. *BioData Mining*, 5:16, 2012. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3605108/.

Marvin N. Wright, Andreas Ziegler, and Inke R. König. Do little interactions get lost in dark random forests? *BMC Bioinformatics*, 17:145, 2016.

**Appendix outline.** For the experiments in Section 3, Appendices A–C describe the benchmark and vary interaction structure, context width and clues for a fixed interaction. Appendices D–F follow the mechanisms studied in Section 4, from interventions on column labels to transfers at the final readout and allocation in forests and xRFM. Appendix G reports the further training discussed in Section 7. The measured track of Section 5 is covered in Appendices H–J: CH65, GPU runtimes, FPS and calibrated tests for omission. Details of the repair in Section 6 appear in Appendices K and L, including residual search, known rules, the guard and unmodified targets. Theory and released code are described in Appendices M and N.

## A Benchmark details

**Data and preprocessing.** The benchmark comprises California housing (8 numeric columns), house_16H (16), protein/CASP (9), diamonds (6), black-friday (5), superconductivity (81), CERN electron collisions (17), King County housing (17), FPS benchmark (27), MSLR/Microsoft (136), NYC taxi (8) and Open-Meteo weather (12). Taxi and weather were collected after the checkpoints were released. We discard categorical and constant columns and replace missing values with medians from training rows. The first five tables permit contexts of both 4,000 and 16,000 rows; we use 4,000 on the others. Every fit reserves 2,000 separate rows for testing. For a given table, sample size and seed, a single generator draws the row subsample, three Gaussian interaction columns, real column permutations and column order. All arms use these draws and the same target. Standardization uses training statistics. We obtain capture by regressing test predictions on $(1, g, y_{\text{real}})$ and taking the coefficient of $g$. The smooth product experiment replaces parity with $0.6z_1z_2z_3$.

**Models.** We use `tabicl` 2.1.1 with checkpoint `v2-20260212` for TabICLv2, `tabpfn` 8.5.0 with eight estimators for TabPFN-3, and `tabfm` 1.0.1 with checkpoint `tabfm-1.0.0` for TabFM. All use package defaults. CatBoost runs for 2,000 iterations, stopping early on 10% of the training rows. We take the tuned default configurations for LightGBM, XGBoost and RealMLP from `pytabkit` 1.7.3. Its defaults are used for random forest, MLP, TabM, FT-Transformer and xRFM (Prokhorenkova et al., 2018; Ke et al., 2017; Chen & Guestrin, 2016; Breiman, 2001; Holzmüller et al., 2024; Gorishniy et al., 2025; 2021; Beaglehole et al., 2025).

**Aggregation.** We first average the three seeds for each table, then bootstrap tables with the arms kept paired. The pooled TFM estimate gives equal weight to all 51 model–table–size cells; a sampled table retains every model and size. Figure 6 reports results for each table at 4,000 rows. We also checked TabPFN-2.5, which behaves like TabPFN-3: crowding is $0.51$ at 16,000 rows and exceeds $0.2$ on each of the five tables.

## B Interaction structure and context width

**Width and order.** For the width experiment, we use the five large tables with 16,000 context rows and seeds 30–32. We fix a random column order within each table and retain its first $m \in \{1, 2, 4, \text{all}\}$ real columns, either aligned or permuted. Every arm retains the full real target. With one, two, four and all real columns, TabPFN-3 loses $0.03$, $0.08$, $0.18$ and $0.42$ in capture (Figure 2a). Joint row permutations preserve dependence among competitors; using seeds 60–62 gives capture $0.78$, compared with $0.77$ for independent column permutations and $0.36$ for aligned columns.

To vary order, we use $k \in \{2, 3, 4\}$, seeds 10–12 and all 12 tables. Retention $\kappa_{\text{aligned}}/\kappa_{\text{perm}}$ for order three is below that for pairs in 14 of the 15 cells where both are learnable. At order four, TabM retains $0.92$ capture while all three TFMs reach the dilution floor.

**Competitor strength.** Increasing the number of informative columns has more effect than increasing their strength. When we retain four columns, choosing those most or least correlated with the target gives crowding of $0.22$ and $0.20$ for TabPFN-3, $0.48$ and $0.48$ for TabICLv2, and $0.30$ and $0.38$ for TabFM. A law fitted to width and order on synthetic tasks describes dilution within that grid, but underpredicts TabPFN-3 crowding with real columns by roughly eightfold.

**Narrow contexts and further tables.** We next give learners two or four real columns from each of the 12 tables, with 4,000 rows and seeds 80–82. We select learnable cells on seed 80 and score

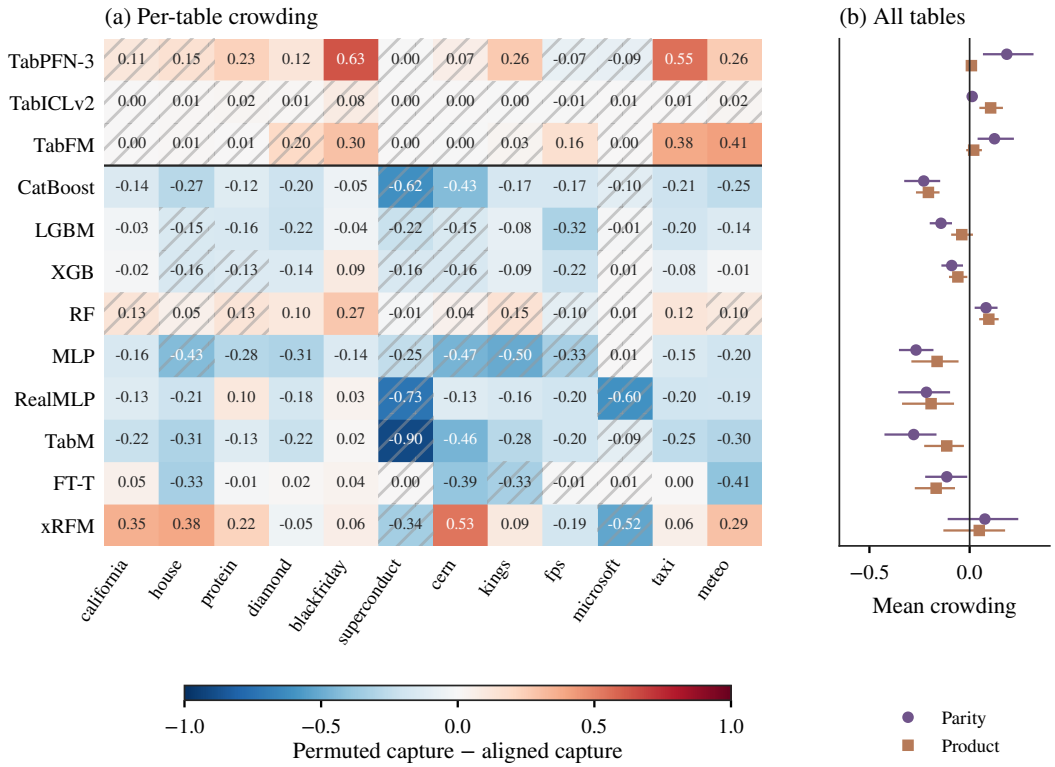


Figure 6: **Results by table distinguish crowding from dilution.** (a) Parity crowding with 4,000 rows. Red denotes suppression and blue improvement; hatching indicates capture below 0.2 in the permuted arm. (b) Mean crowding over all tables for parity and smooth products. The 95% intervals bootstrap tables.

them on seeds 81–82. Mean crowding is 0.30 (24 cells), 0.40 (16) and 0.32 (17) for TabPFN-3, TabICLv2 and TabFM. Applying the same selection at full width gives 0.31 $[0.19, 0.42]$ over 17 cells; changing the learnability threshold from 0.1 to 0.4 gives 0.30–0.34. We also evaluate 22 additional public numeric tables with up to 4,000 context rows. Pairwise crowding for the three TFMs is 0.04 $[0.01, 0.07]$, 0.19 $[0.11, 0.26]$ and 0.34 $[0.23, 0.46]$; for order three it is 0.30 $[0.12, 0.48]$, 0.31 $[0.08, 0.53]$ and 0.21 $[0.03, 0.41]$. CatBoost and TabM benefit at both orders.

**Classification and genotype coding.** For binary classification, we predict $\mathbf{1}[g + y_{\text{real}} + e > 0]$ on five tables with 8,000 rows and seeds 50–52. We score capture on predicted log odds relative to the fit using only planted columns. Among learnable interactions, crowding is 0.28 for TabPFN-3 (five tables), 0.28 for TabICLv2 (one table) and 0.52 for TabFM (three tables); CatBoost and TabM gain. We also simulate a quantitative trait with 20 SNPs. Under dominance coding, TabICLv2 and TabFM strongly crowd three-way epistasis in some regimes. Effects are weaker with additive coding and for TabPFN-3. TabM crowds in two settings with dominance coding (0.17 and 0.30).

**Rule families.** We compare nine rules on the width design, using seeds 70–71 and scaling each rule to standard deviation 0.6. Their simpler structure is measured by the held-out $R^2$ of an 8-bin additive and $4 \times 4$-bin pairwise fit. When this fit explains at most 6% of a rule, crowding ranges from 0.23–0.56. It is 0.01–0.19 at 24%, and at most 0.04 for random depth-3 trees, AND and majority rules (64–85%). A parity built from three real columns inherits simpler structure through input dependence (8–93% explained). Models capture 0.94–1.01 in both arms. With independent support columns, crowding remains at amplitudes 0.3, 0.6 and 1.2. In Figure 2c we scale the blend $g \propto (1 - \alpha)\,\text{sign}(z_1 z_2 z_3) + \alpha z_1 z_2 z_3$ to standard deviation 0.6, using $\alpha \in \{0, .25, .5, .75, 1\}$, five tables, 16,000 rows and seeds 50–51. TabPFN-3 receives every real column; the others receive two, preserving parity learnability in the permuted arm.

**Uninformative columns and context size.** We record the first tested context size, from 250 to 16,000 rows, at which mean capture reaches 0.5 (Figure 7). For TabICLv2, parity never crosses this threshold with 11 or more columns. The smooth product remains learnable at every width.

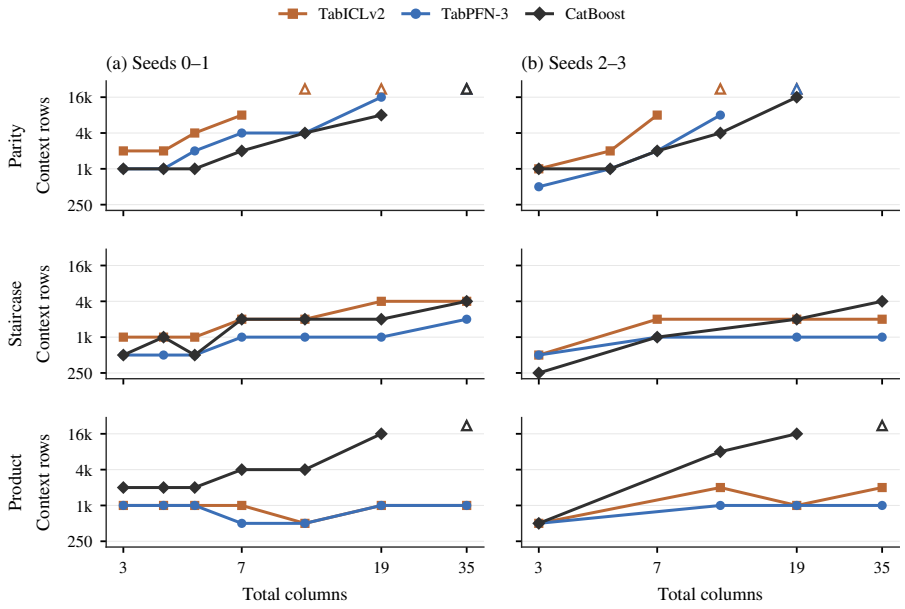


Figure 7: **Uninformative columns increase the context needed to learn an interaction.** First tested context size with capture at least 0.5. Open triangles indicate no crossing by 16,000 rows. Columns correspond to two seed pairs and rows to parity, staircase and smooth product targets.

## C Fixed-interaction clues

We add a clue on the same three Gaussian inputs as the parity, whose amplitude remains 0.6. This additive or pairwise clue is $L^2$-orthogonal to the parity; hidden versions instead use fresh, unobserved inputs of the same distribution. We permute the four real competing columns jointly within context and query rows, keeping labels and column marginals identical between arms. California housing, house_16H and diamonds supply the real tables. We use seeds 9200–9202, contexts of 4,000 or 16,000 rows and 2,000 query rows. Capture comes from regressing predictions on the intercept, real outcome, clue and $g$.

TabPFN-3 and TabICLv2 use eight estimators; TabFM uses its regression recipe with 32 estimators. CatBoost has 1,000 depth-6 trees, learning rate 0.05 and leaf penalty 3, while TabM uses defaults. A pilot on two tables set the grid and is excluded. Figure 8 includes all 1,980 fits. Crowding peaks at intermediate clue strengths for TabICLv2 and TabFM: weak clues first make parity learnable with permuted columns (Figure 3), then stronger clues reduce the gap by improving aligned capture.

## D Column-label interventions

**Tasks and hooks.** Synthetic tasks contain three Gaussian parity columns with $g = 0.6\,\mathrm{sign}(z_1 z_2 z_3)$, $K$ Gaussian competitors and observation noise. The informative arm adds $0.3\sum_j x_j^{\mathrm{comp}}$. The hidden arm replaces that term with independent Gaussian nuisance of equal variance, and the noise arm omits it. Each task uses 16,000 context rows, 2,000 test rows and ten seeds. For TabPFN-3 with real columns, we compare joint row permutation with alignment on California, house_16H and protein, using four seeds each. Hooks substitute labels at TabICLv2's `y_encoder`, TabPFN-3's `col_y_encoder`, or TabFM's cell embedder. An affine transformation matches them to native labels. Every forward call checks row order, the affine map and the unchanged prediction la-

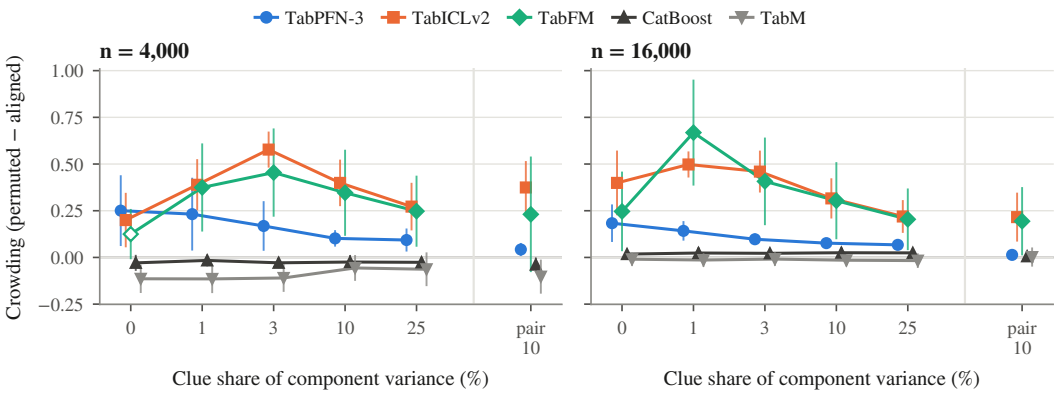


Figure 8: **Adding clues to a fixed interaction.** Capture in the permuted arm minus aligned capture, with 95% intervals over nine table–seed units. We vary visible additive clues and include a pairwise clue at 10%. Hollow markers indicate capture below 0.2 in the permuted arm. Hiding clues of the same distribution changes crowding by at most 0.034 from the value without clues, for every learner. CatBoost and TabM show no crowding at any strength.

bels, with discrepancies below $5 \times 10^{-7}$. TabPFN-3's target power transform is disabled throughout the comparison.

**Controls.** Our control changes column labels with a signal of equal variance that is independent of observed columns: a permuted competitor term on synthetic tasks and a permuted real signal on real tables. These perturbations have much smaller effects than the true swaps (Table 3). Shuffling column labels also releases crowding for TabICLv2 and TabPFN-3. Setting label embeddings to zero is harder to interpret. For TabPFN-3 it lowers capture with permuted columns from 0.72 to 0.46, leaving aligned capture nearly unchanged ($0.18 \to 0.19$); for TabFM it eliminates capture in both arms.

Table 3: **Label swaps and controls of equal variance.** Capture using native column labels, true swaps, and control perturbations independent of observed columns. Labels at the prediction stage are fixed. Intervals bootstrap ten synthetic tasks or twelve real-table/seed units.

| Model | Arm | units | own labels | real swap | control | control − own |
|---|---|---|---|---|---|---|
| TabPFN-3 | aligned (real) | 12 | 0.18 [0.11, 0.25] | 0.60 [0.49, 0.69] | 0.23 [0.18, 0.28] | 0.05 [0.03, 0.08] |
| TabPFN-3 | rows permuted (real) | 12 | 0.72 [0.64, 0.80] | 0.41 [0.23, 0.57] | 0.73 [0.63, 0.81] | 0.00 [-0.02, 0.03] |
| TabICLv2 | informative | 10 | 0.08 [0.04, 0.13] | 0.65 [0.61, 0.68] | 0.07 [0.04, 0.11] | -0.01 [-0.02, 0.02] |
| TabICLv2 | uninformative | 10 | 0.71 [0.68, 0.74] | 0.08 [0.04, 0.13] | 0.72 [0.69, 0.74] | 0.01 [-0.01, 0.02] |
| TabFM | informative | 10 | 0.33 [0.21, 0.45] | 0.78 [0.70, 0.84] | 0.34 [0.22, 0.46] | 0.01 [-0.04, 0.06] |
| TabFM | uninformative | 10 | 0.80 [0.73, 0.85] | 0.32 [0.21, 0.45] | 0.78 [0.72, 0.84] | -0.02 [-0.05, 0.01] |

**A nonmonotone interaction.** We also use $b(z) = \mathbf{21}[|z| > \Phi^{-1}(.75)] - 1$ and $g = .6\,\mathrm{sign}(z_1)\,\mathrm{sign}(z_2)b(z_3)$. All three bits are balanced, so $g$ is independent of every proper input subset. Its even third factor also makes it orthogonal to products of odd sign and rank transforms. For each model we draw eight tasks with 16,000 context and 4,000 query rows. TabICLv2 and TabFM receive two linear competitors; TabPFN-3 receives eight squared competitors. Capture with hidden nuisance minus capture with informative competitors is 0.59 $[0.55, 0.61]$, 0.38 $[0.25, 0.48]$ and 0.23 $[0.09, 0.37]$ for TabICLv2, TabFM and TabPFN-3. Swapping labels releases and induces this gap (Table 4).

**Representations and token groups.** To probe TabICLv2's query representation, we use five tasks with 16,000 context and 8,000 query rows. We reduce the 512 dimensions with 32-component PCA and fit a two-layer MLP on half the queries. It recovers parity with $R^2$ 0.53 under informative columns and 0.67 after swapping labels, versus 0.95 from raw support columns. Ridge reaches only 0.01 and 0.04, suggesting that much of the interaction remains in nonlinear form.

Table 4: **A nonmonotone interaction.** Capture averaged over eight tasks. Floor counts tasks with capture below 0.2 in the noise arm. Both swap columns change labels only at the column stage.

| Model | support only | noise | hidden | informative | info. swap | noise swap | floor |
|---|---|---|---|---|---|---|---|
| TabPFN-3 | 0.935 | 0.249 | 0.231 | 0.002 | 0.134 | 0.000 | 6/8 |
| TabICLv2 | 0.784 | 0.620 | 0.585 | -0.001 | 0.568 | 0.000 | 0/8 |
| TabFM | 0.931 | 0.626 | 0.541 | 0.164 | 0.586 | 0.172 | 0/8 |

We also transplant representations of column groups from a donor with swapped labels on ten tasks and all eight ensemble members. Transferring all groups reproduces the donor, raising capture from 0.045 to 0.668. Transferring only groups containing the complete support gives 0.040: release requires broader coverage.

## E Routing and representations at the final readout

At the last feature aggregation block of TabPFN-3, the four CLS queries compute

$$H = R + W_o(AV), \qquad A = \mathrm{softmax}(QK^\top/\sqrt{d}), \qquad \mathrm{out} = H + \mathrm{MLP}(\mathrm{LN}(H)).$$

We transplant routing probabilities $A$ through projected $Q, K$, projected values $V$, and query skip $R$, individually and in combination. The output projection, normalization, MLP and subsequent layers remain native. Donor passes use swapped column labels with matched mean and variance, and no target power transform. For release, independent Gaussian nuisance replaces the real target component; for induction, that component is aligned with the permuted columns. We test parity, the nonmonotone rule from Appendix D, and $0.6z_1z_2z_3$. Each family has twelve table–seed pairs from California, house_16H and protein, with 16,000 context rows, 2,000 query rows and eight estimators. Empty transfers give exactly the native predictions, and full transfers reproduce the donor.

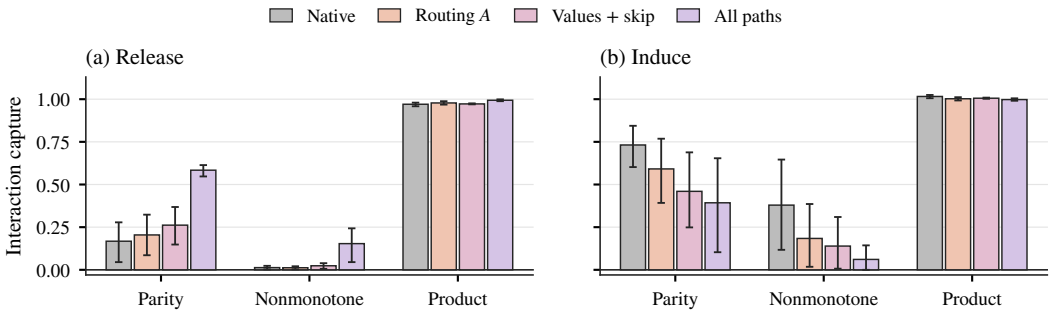


Figure 9: **Attention routing alone restores little of the missing interaction.** TabPFN-3 capture after receiving routing $A$, representations $V + R$, or all three from a donor with swapped column labels. Bars average four seeds for each of three tables; intervals bootstrap tables.

**Routing has a larger effect with donor values and skip.** With native values and skip, donor routing restores only 0.037 of parity capture. With donor values and skip, it restores 0.322. We see the same dependence for the nonmonotone rule ($-0.001$ and 0.130), while the smooth product changes little. Routing can thus matter even when its isolated effect is small. Control donors follow random directions matched to the true donor's mean, variance, correlation with native labels and displacement. They produce much smaller effects: full parity release is 0.416 for the true donor and 0.079 for the random donor. For the nonmonotone rule, true release and induction are 0.141 and $-0.318$, versus $-0.008$ and $-0.035$ for random donors. Full transfer also reduces capture of the easy signal ($0.776 \to 0.672$ on parity); this identifies a pathway but is not itself a correction.

Table 5: **Transfers at the final readout.** Mean capture across three tables and four seeds per family. Release reverses the label substitution used for induction. Transferring all paths reproduces the donor.

| | Median parity | | Nonmonotone | | Smooth product | |
|---|---|---|---|---|---|---|
| Donor paths | Release | Induction | Release | Induction | Release | Induction |
| Native | 0.17 | 0.73 | 0.01 | 0.38 | 0.97 | 1.02 |
| Routing $A$ | 0.20 | 0.59 | 0.01 | 0.18 | 0.98 | 1.00 |
| Values $V$ | 0.15 | 0.61 | 0.01 | 0.23 | 0.96 | 1.02 |
| $A + V$ | 0.32 | 0.38 | 0.02 | 0.10 | 0.98 | 1.00 |
| Skip $R$ | 0.16 | 0.54 | 0.02 | 0.17 | 0.97 | 1.00 |
| $A + R$ | 0.41 | 0.35 | 0.06 | 0.07 | 1.00 | 0.99 |
| $V + R$ | 0.26 | 0.46 | 0.02 | 0.14 | 0.97 | 1.01 |
| All $A + V + R$ | 0.58 | 0.39 | 0.15 | 0.06 | 0.99 | 1.00 |

## F EXPLICIT FEATURE ALLOCATION IN FORESTS AND XRFM

We reuse the tables, rows, seeds and arms of Appendix A. Forests have 300 trees and consider all columns, $\sqrt{p}$ columns, or one random column per split. ExtraTrees additionally samples thresholds, making its structure independent of labels in the one-candidate setting. The residual forest fits a second forest to predictions' residuals obtained out of fold with two folds. xRFM either runs its default five AGOP iterations or keeps an identity metric (Table 6).

We measure support allocation by its share of splits, weighted by sample count, or its share of diagonal AGOP weight, relative to the uniform share $3/p$. Informative competitors reduce xRFM support weight from $3.65$ to $0.90$ times uniform at 16,000 rows. Across 17 table–size cells the loss tracks crowding, with Spearman $0.96$ for xRFM and $0.67$ for forests (Figure 10). Forests with one candidate and ExtraTrees reach the dilution floor.

Table 6: **Allocation variants with 16,000 context rows** on five tables. Floor counts tables with capture below $0.2$ in the permuted arm. Support allocation is relative to uniform, permuted $\rightarrow$ aligned. With 4,000 rows, forests reach the floor on 8–12 of 12 tables; xRFM crowding is $+0.08$ with AGOP and $-0.08$ without.

| Learner | NONE | PERM | ALIGNED | crowding [95% CI] | floor | allocation |
|---|---|---|---|---|---|---|
| RF (benchmark configuration) | 0.96 | 0.41 | 0.20 | $+0.21$ $[+0.12, +0.30]$ | 1/5 | – |
| RF, all candidate columns | 0.96 | 0.44 | 0.21 | $+0.23$ $[+0.15, +0.30]$ | 1/5 | $1.39 \rightarrow 0.94$ |
| RF, $\sqrt{p}$ candidates | 0.89 | 0.26 | 0.11 | $+0.15$ $[+0.12, +0.18]$ | 3/5 | $1.23 \rightarrow 0.84$ |
| RF, one random candidate | 0.89 | 0.12 | 0.09 | $+0.03$ $[+0.01, +0.04]$ | 4/5 | $1.05 \rightarrow 1.07$ |
| ExtraTrees, one candidate | 0.68 | 0.08 | 0.09 | $-0.01$ $[-0.02, 0.00]$ | 5/5 | $1.05 \rightarrow 1.06$ |
| RF + residual refit | 1.00 | 0.57 | 0.46 | $+0.10$ $[-0.01, +0.24]$ | 0/5 | – |
| xRFM, AGOP on (default) | 0.89 | 0.78 | 0.59 | $+0.20$ $[+0.07, +0.31]$ | 0/5 | $3.65 \rightarrow 0.90$ |
| xRFM, AGOP off | 0.89 | 0.43 | 0.52 | $-0.08$ $[-0.14, -0.04]$ | 0/5 | – |

For controlled swaps we use $K = 2, 4$ Gaussian competitors, 16,000 rows and seeds 500–509. A forest grows its splits using one set of labels and fits leaf values with another. Removing the competitor from split labels raises capture from $0.32$ to $0.78$; the reverse swap lowers it from $0.78$ to $0.32$. Release also reduces capture of the easy signal ($0.98 \rightarrow 0.76$ at $K = 2$). Gaussian competitors induce little crowding in RFM, and metric swaps change its capture by only $0.02$ and $0.07$.

## G CONTINUED TRAINING

Label swaps locate a pathway (Section 4), but leave open whether architecture or learned weights determine crowding. We continue training the released TabICLv2 weights while retaining its architecture and public pretraining generator, with no tasks designed for parity. This experiment concerns one model, classification and a fixed training budget. We assess reference capture and ordinary-table log loss alongside suppression.

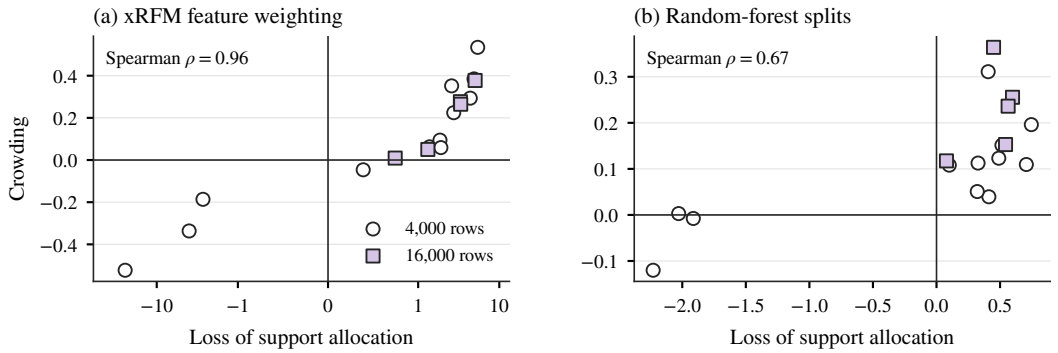


Figure 10: **Lower support allocation accompanies greater crowding.** All 17 table–size cells, with allocation relative to uniform. The horizontal axis measures permuted minus aligned allocation; the vertical axis measures permuted minus aligned capture. Hollow markers denote 4,000 rows and filled markers 16,000. The xRFM horizontal axis uses a symmetric log scale.

**Training.** Starting from `tabicl-classifier-v2-20260212`, we train six streams with seeds 100–105 for 2,000 updates on the public generator of mixed structural causal models. Every update uses 64 tables, each with 12 columns and 512 rows. We run the first 2,000 steps of an 8,000-step schedule: 400 warm-up steps followed by cosine decay. AdamW uses learning rate $10^{-5}$, no weight decay, bfloat16 and gradient clipping at 1. Loading the released weights through this pipeline gives exactly the released class probabilities.

**Evaluation.** We use the equal variance design of Figure 4a, binarizing targets at the context median. Each task has 16,000 context rows and 2,000 query rows. We regress predicted probabilities on $(1, g, c)$ and divide the coefficient of $g$ by its counterpart for the Bayes probability to obtain capture. Intervals use a crossed bootstrap over streams and tasks.

**Results.** Across 20 fresh task seeds and $K = 2, 4$, capture with informative columns rises from $0.061$ to $0.567$, gaining $0.505$ $[0.444, 0.559]$. Log loss improves by $0.039$ (Table 7). Costs in reference arms are smaller and concentrated at $K = 4$: capture with noise columns changes from $0.733$ to $0.626$, and with hidden nuisance from $0.704$ to $0.561$. The median log loss ratio on 31 ordinary tables is $1.0065$; within each stream, $9.7$–$19.4\%$ of tables worsen by more than $5\%$. One stream retains a $0.159$ gap at $K = 4$, just over the $0.15$ target. Without further selection, training also helps other families (Table 8). Gains are strong for smooth products and parity, partial for the nonmonotone rule at $K = 2$, and weak for all-even and modular rules that the released model already misses without competitors. Unlike in regression, the released classifier strongly crowds the smooth product: binarization at the median removes its variance cue.

Table 7: **Further training removes most suppression.** Parity capture for released TabICLv2 and the mean of six further training streams, using 20 fresh task seeds per $K$. The 95% intervals use a crossed bootstrap over streams and tasks.

| Quantity | $K = 2$ | $K = 4$ |
|---|---|---|
| Informative capture gain | $0.562$ $[0.492, 0.622]$ | $0.449$ $[0.380, 0.513]$ |
| Suppression gap after training | $0.059$ $[0.039, 0.075]$ | $0.059$ $[0.003, 0.118]$ |
| Suppression-gap reduction | $0.601$ $[0.546, 0.655]$ | $0.624$ $[0.568, 0.680]$ |
| Informative log-loss benefit | $0.039$ $[0.032, 0.045]$ | $0.039$ $[0.035, 0.042]$ |
| Noise-arm capture change | $-0.039$ $[-0.119, 0.019]$ | $-0.175$ $[-0.283, -0.074]$ |
| Hidden-arm capture change | $-0.043$ $[-0.129, 0.022]$ | $-0.243$ $[-0.349, -0.133]$ |

Table 8: **Transfer to other interaction families.** Capture with noise or informative columns for the released TabICLv2 classifier and the mean of six further training streams. Gains with informative columns have 95% intervals from crossed bootstrap resampling of streams and ten task seeds. We use $e(z) = 2\,\mathbf{1}\{|z| > \Phi^{-1}(0.75)\} - 1$, the mixed rule $\operatorname{sign}(z_1)\operatorname{sign}(z_2)e(z_3)$, and the all-even rule $\prod_j e(z_j)$.

| | | Released | | Continued training | | |
|---|---|---|---|---|---|---|
| Family | $K$ | Noise | Informative | Noise | Informative | Informative gain |
| Parity | 2 | 0.75 | 0.19 | 0.74 | 0.70 | 0.51 [0.42, 0.59] |
| Parity | 4 | 0.68 | 0.00 | 0.56 | 0.48 | 0.48 [0.41, 0.55] |
| Mixed nonmonotone | 2 | 0.73 | 0.01 | 0.45 | 0.25 | 0.25 [0.15, 0.35] |
| Mixed nonmonotone | 4 | 0.43 | 0.00 | 0.12 | 0.04 | 0.04 [0.02, 0.07] |
| All-even | 2 | 0.11 | 0.00 | 0.02 | 0.01 | 0.01 [0.00, 0.01] |
| All-even | 4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 [0.00, 0.00] |
| Three-state modular | 2 | 0.58 | 0.00 | 0.31 | 0.12 | 0.12 [0.07, 0.19] |
| Three-state modular | 4 | 0.17 | 0.00 | 0.05 | 0.02 | 0.02 [0.01, 0.04] |
| Smooth product | 2 | 0.95 | 0.77 | 1.01 | 1.01 | 0.24 [0.19, 0.31] |
| Smooth product | 4 | 0.95 | 0.07 | 0.97 | 0.98 | 0.90 [0.83, 0.97] |

## H CH65 ANTIBODY BINDING

**Data and summaries.** We take CH65 affinities from the authors' data after their quality and replicate filtering, using mean $-\log_{10} K_D$ for MA90, G189E and SI06 (Phillips et al., 2023). Each assay has 16 binary mutation indicators. About 8,000 reference rows select the largest coefficient of order three in a joint polynomial fit and fit ridge summaries with penalty 10. The *full* summary includes all main effects and pairs. Complementary summaries use only inputs outside the selected triple: a ridge on all remaining inputs (*comp. all*), four ridges on disjoint groups (*groups*), and their sum (*group sum*). None has a coefficient on the scored product.

**Evaluation scope.** Exploratory development selected CH65, a context of 512 rows and the complementary summaries. Evaluation rows had not been used previously for the same assay, although most sequences had appeared in earlier fits with outcomes from another antigen. Only reference rows select the scored component. Our conclusions are conditional on these choices and apply to this selected system.

**Design and scoring.** We append the summary features to the 16 native inputs, either aligned or jointly permuted within context and query rows. TabPFN-3 and TabICLv2 use eight estimators, TabFM uses 32, and CatBoost follows the benchmark recipe. Each of 24 seeds draws 512 context rows and 2,048 query rows. After projecting out all main effects and pairs, we divide the held-out coefficient of the selected product in predictions by its coefficient in the measured outcome to obtain capture. Paired seed bootstraps and paired-$t$ intervals use Bonferroni adjustment over 36 TFM contrasts, and separately over 12 CatBoost contrasts. Suppression requires both adjusted intervals to exclude zero, aligned capture below $0.95$ and permuted capture at least $0.2$.

**Results.** We report all 60 contrasts in Table 9. The full summary lowers measured G189E capture for all five learners; TabPFN-3 also loses capture on SI06. TabFM loses $0.156$ on G189E: its adjusted $t$ interval $[0.015, 0.297]$ passes, while the bootstrap interval $[-0.002, 0.240]$ narrowly fails. Further control for every other product of order three gives G189E losses of $0.191$, $0.164$ and $0.164$ across the TFMs. TabM was added later on the same tasks, with a separate adjustment over its 12 contrasts. On G189E it loses $0.33$ (adjusted bootstrap $[0.19, 0.55]$, $t$ $[0.12, 0.54]$; all 24 seeds), even as RMSE improves $10.8\%$. No other TabM contrast has significant loss. Most summaries improve capture on MA90, with one significant exception: grouped complementary summaries for TabICLv2. In 12 further seeds with nested contexts (Figure 11), summaries often help at 128 rows, suppress capture at 512, and have little effect at 2,048, where both arms approach unit capture on G189E and SI06. Fitting the target minus the summary and adding the summary back rarely helps both capture and RMSE, and never does for TabFM in any assay.

Table 9: **All CH65 contrasts.** Capture with permuted summaries minus capture with aligned summaries, using 24 seeds and 512 context rows. Bold entries indicate adjusted evidence of suppression; TabFM on G189E passes only the $t$ interval. TabM was added later, with adjustment over its own 12 contrasts as for CatBoost.

| Assay | Model | full | comp. all | groups | group sum |
|---|---|---|---|---|---|
| G189E | TabPFN-3 | **+0.18** | −0.02 | +0.01 | +0.00 |
| G189E | TabICLv2 | **+0.15** | −0.06 | +0.02 | −0.01 |
| G189E | TabFM | **+0.16** | −0.11 | −0.11 | −0.13 |
| G189E | CatBoost | **+0.13** | −0.14 | −0.08 | −0.12 |
| G189E | TabM | **+0.33** | −0.09 | −0.04 | −0.08 |
| MA90 | TabPFN-3 | −0.23 | −0.29 | −0.03 | +0.02 |
| MA90 | TabICLv2 | −0.13 | −0.33 | **+0.12** | **+0.10** |
| MA90 | TabFM | −0.41 | −0.46 | −0.13 | −0.15 |
| MA90 | CatBoost | +0.07 | −0.05 | −0.07 | −0.01 |
| MA90 | TabM | +0.12 | −0.21 | −0.07 | −0.12 |
| SI06 | TabPFN-3 | **+0.08** | +0.00 | +0.06 | +0.01 |
| SI06 | TabICLv2 | −0.02 | −0.03 | +0.05 | −0.03 |
| SI06 | TabFM | +0.04 | −0.02 | −0.04 | −0.03 |
| SI06 | CatBoost | −0.07 | −0.09 | −0.10 | −0.07 |
| SI06 | TabM | +0.02 | −0.05 | −0.07 | −0.05 |

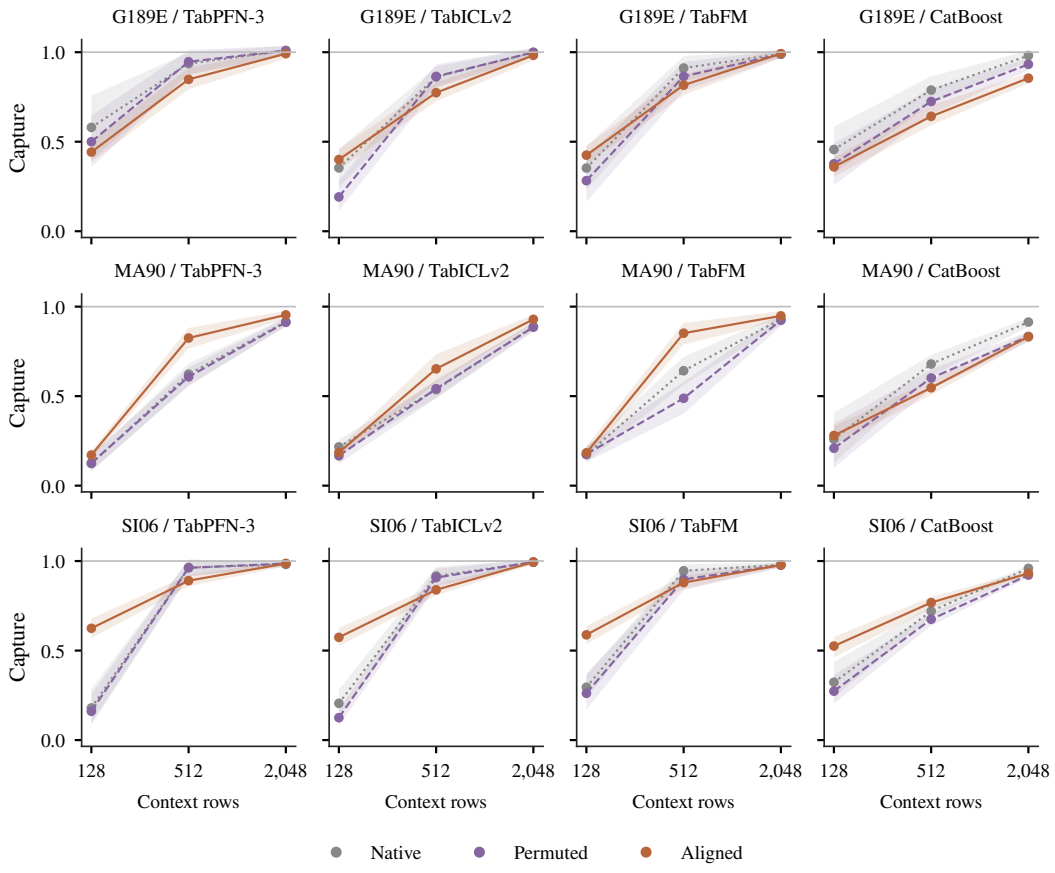


Figure 11: **CH65 results vary with context size.** Capture from native inputs and from adding the full pairwise summary, permuted or aligned. Each of 12 seeds shares query rows across nested contexts and models. Bands give 95% intervals from bootstrapping seeds.

# I GPU RUNTIMES AND FPS

**SGEMM.** SGEMM records four timings of a $2048 \times 2048$ matrix product on one GPU for each of 241,600 feasible configurations of 14 discrete kernel parameters (Paredes & Ballester-Ripoll, 2017). We predict $\log_{10}$ of mean runtime. A fixed partition assigns 16,384 rows to fit a ridge summary of main effects and pairs, with penalty 10 and a design of 336 columns, rank 322. An earlier study screened this summary on another 16,384 rows, then confirmed it on twelve disjoint blocks of 2,048 context and 8,192 query rows. At 2,048 context rows, RMSE rose $4.5\%$ $[3.1, 6.0]$ for TabPFN-3 and $8.3\%$ $[7.1, 9.5]$ for TabICLv2, while CatBoost improved. A prespecified component on three inputs gave mixed results. For the main text we fix error endpoints in advance and use eight new disjoint blocks, each with 2,048 context and 2,048 query rows, excluding all previously used configurations. Contexts of 128 rows are nested within these blocks.

**KTT.** The Kernel Tuning Toolkit (Hozzová et al., 2021) supplies ten standard combinations of kernels and devices on GTX 1070 and RTX 2080. Our rule includes all cells with at least 3,000 configurations and no more than 256 basis columns for main effects and pairs. This admits GEMM (5,788 and 5,786 configurations; 164 basis columns) and $N$-body (3,084 and 3,088; 183 columns). Convolution exceeds the basis limit; transpose and Coulomb have too few rows. Predictors receive only tuning parameters. For each cell we reserve 768 reference rows and 1,024 fixed query rows, drawing six contexts from the remainder. The target is log measured duration.

**Endpoints.** The five learners follow Appendix C. We split prediction errors into their projection on the query design's categorical main effects and pairs and an orthogonal remainder. Their mean squares sum to total MSE; the remainder's mean square is termed beyond-pair error and includes measurement noise. At the longer context, one-sided paired-$t$ tests on log ratios test RMSE harm above 3% and any increase in beyond-pair MSE. Holm adjustment covers six SGEMM tests and 24 KTT tests: three TFMs, two endpoints and four cells. Figure 12 includes every comparison at the longer context.

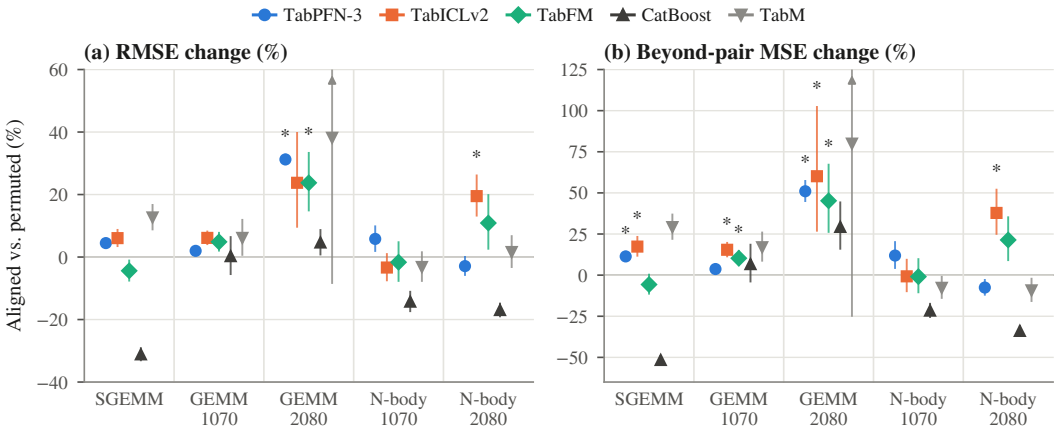


Figure 12: **Runtime comparisons at longer contexts.** Changes from permuted to aligned summaries in (a) RMSE and (b) beyond-pair MSE, with 95% paired-$t$ intervals. SGEMM uses 2,048 context rows and eight blocks; KTT uses 1,024 rows and six contexts. Asterisks denote Holm-adjusted primary tests for TFMs. Arrows indicate intervals beyond the axes. At 128 rows, summaries improve RMSE for every learner and setting.

**A measured column-label intervention.** We reserve four further SGEMM blocks with 2,048 context rows and fit TabPFN-3 and TabICLv2 once for each input arm. Predictions use intact column labels, an identity substitution, the target minus the summary, or a random direction matched to the residual labels' mean, variance, correlation and displacement. We change only the input to the column label encoder; prediction labels and inputs are identical bit for bit. Residual labels worsen beyond-pair MSE in every arm (Table 10), including $17.5\%$ for TabPFN-3 and $27.2\%$ for TabICLv2

with aligned inputs. The smaller aligned/permuted gap reflects deterioration in permuted predictions rather than recovery.

Table 10: **Residual column labels on SGEMM.** Percentage change in beyond-pair MSE when residual labels replace intact labels or a matched random direction. Intervals cover 95% over four blocks.

| Model | Input arm | Residual vs. intact | Residual vs. random |
|---|---|---|---|
| TabPFN-3 | native | 29.3 [20.8, 38.3] | 12.3 [3.5, 21.8] |
| TabPFN-3 | permuted | 32.7 [19.7, 47.1] | 13.8 [4.1, 24.5] |
| TabPFN-3 | aligned | 17.5 [15.6, 19.3] | 8.0 [4.8, 11.2] |
| TabICLv2 | native | 53.7 [32.7, 78.1] | 1.1 [−15.0, 20.4] |
| TabICLv2 | permuted | 63.7 [33.8, 100.4] | −2.3 [−32.2, 40.8] |
| TabICLv2 | aligned | 27.2 [17.6, 37.5] | −6.8 [−16.1, 3.6] |

**FPS.** FPS records video game frame rates for hardware and software configurations (Peeters et al., 2021) (OpenML 44992). We screened 31 families outside biology using two seeds and three models. Depth-2 CatBoost teachers, with 1,000 trees, learning rate 0.03 and leaf penalty 5, supplied one full summary and four grouped summaries. Contrasts advanced only if both seeds showed at least 3% harm. Eight contrasts across five families passed; we evaluated each family with all models and summaries on eight new contexts of 512 rows (Table 11). FPS, CERN, Pol and Seattle confirm, but diamonds does not. Grouped FPS summaries raise TabPFN-3 RMSE by 22.7% [18.8, 27.5] over permutation and 29.5% over native inputs, while helping CatBoost. TabFM and TabM, added later on the same contexts, improve by 4.2% [1.2, 7.2] and 76.1%. TabM fails to fit native FPS at 512 rows (RMSE 0.83, versus 0.049 for TabFM), so every summary helps it. On other families, TabFM shares the CERN harms of the other TFMs (+17.6% and +31.3%), while TabM changes by at most +2.0% (Table 11). Diamonds effects shrink to 0.4–0.6%, with intervals including zero. CERN harms are larger, but the supplied kinematics determine its target; Seattle nearly duplicates training and test features, and Pol has poor source documentation. Over all 31 families, full summaries improve average RMSE 1.9% for TabPFN-3 and 2.9% for TabICLv2. Grouped summaries cost TabPFN-3 1.9%.

Table 11: **Confirmation on five promoted families.** Percentage RMSE change from permuted to aligned summaries across eight contexts; negative values indicate improvement. *Contrasts advanced from the screen. TabFM and TabM were added later on these contexts and did not participate in screening. ‡TabM barely fits native FPS at 512 rows.

| | TabPFN-3 | | TabICLv2 | | TabFM | | CatBoost | | TabM | |
|---|---|---|---|---|---|---|---|---|---|---|
| Family | full | groups | full | groups | full | groups | full | groups | full | groups |
| CERN | +29.7* | +43.8* | +10.2 | +30.4* | +17.6 | +31.3 | −12.3 | −2.0 | +2.0 | +2.0 |
| Diamonds | −5.3 | +0.6* | −11.3 | +0.4* | −3.1 | +1.9 | −9.8 | −8.0 | −38.2 | −27.4 |
| FPS | −0.2 | +22.7* | −11.2 | +5.3 | −26.5 | −4.2 | −58.3 | −55.6 | −74.6‡ | −76.1‡ |
| Pol | +1.1 | +8.7* | −15.4 | −2.1 | −15.2 | +2.3 | −16.5 | +4.4 | −28.8 | −27.2 |
| Seattle | −0.4 | −0.3 | −0.1 | −0.2 | −0.6 | −0.9 | +2.1 | +3.6* | −2.5 | −2.0 |

**Correction at equal label budgets.** For residual fitting with a summary, we retain the full summary as an input, fit the target minus the summary, and add it back to predictions. This uses the same 2,512 reference and context labels as pooled native fitting. On FPS it improves RMSE in all eight contexts, by 13.9% [11.1, 16.7] for TabPFN-3 and 14.9% [11.7, 18.3] for TabICLv2 (Table 12). The correction is not uniformly helpful: it harms CERN and log-runtime SGEMM. The guard in Appendix K.3 removes these harms.

Table 12: **Correction with equal label budgets.** Percentage RMSE change for residual fitting with a summary versus pooled fitting on native inputs using the same labels. Negative values denote improvements.

| Task | TabPFN-3 | TabICLv2 | CatBoost |
|---|---|---|---|
| CERN | +39.5 | +88.3 | +12.4 |
| Diamonds | +2.3 | −0.9 | −2.0 |
| FPS | −13.9 | −14.9 | −47.5 |
| Pol | +18.6 | −4.8 | −11.1 |
| Seattle | +0.7 | +0.7 | +0.2 |
| SGEMM (log) | +17.3 | +26.8 | −25.3 |
| SGEMM (raw) | −29.9 | +29.6 | −50.5 |

## J CALIBRATED OMISSION TEST ON DISCRETE TABLES

Discrete inputs give a finite class of main and pairwise effects that we can remove exactly. We use this to test whether a model's held-out residual retains a pure third-order component, without planting any signal. Such a component occurs in the target, has no main or pairwise structure, and is unused by the model.

**Why pure components matter.** We define purity relative to an input distribution and a function class. Removing main effects and pairs leaves a component outside that class (Lengerich et al., 2020), although the same inputs can be individually predictive through other parts of the target. This orthogonality is weaker than the independence of parity from every proper input subset. By itself it establishes neither crowding nor prevalence. CrowdBench first uses parity to isolate the failure, then examines clues and measured targets to determine its scope.

**Evidence from measured systems.** In a complete fluorescent protein landscape, Poelwijk et al. (2019) find terms of higher order averaged over backgrounds. For six software systems with configurable options, Siegmund et al. (2015) explain measured performance using effects of individual options and their interactions. Those interactions often form a hierarchy with clues of lower order, motivating our tests of clues for a fixed interaction. Both studies support mixed targets; showing that a learner suppresses a component still requires the matched interventions in Appendices H–I. Response scale also matters. Across 20 protein datasets, main effects and pairs with a nonlinear link explain a median 96% of phenotypic variance (Park et al., 2024). Interactions of higher order may therefore form only a small remainder of the target. Measuring whether a learner retains that remainder complements average accuracy. Our test quantifies it on the measured scale, without establishing its biological mechanism or why it was omitted.

**Cohort and test.** We study nine measured discrete targets: CH65 G189E, MA90 and SI06 (Phillips et al., 2023); CR9114 H1 and H3 (Phillips et al., 2021); eqFP fluorescence (Poelwijk et al., 2019); HDV and ligase ribozyme activity (Bendixsen et al., 2019); and SGEMM log runtime. Seven UCI nominal tables (Dua & Graff, 2019) add car, nursery, kr-vs-kp, tic-tac-toe, connect-4, mushroom and chess-krkopt, scored as majority class versus rest. We use GAMETES three-way epistasis (Urbanowicz et al., 2012) as a positive control. A search split $S$ has $\min(4{,}000, n/2)$ rows and an evaluation split $E$ up to 16,000 rows. On $S$, residuals from two-fold out-of-fold predictions select one effect-coded contrast on three inputs after removing the intercept, main effects and pairs within the triple. On $E$, we regress residuals of the model fitted on $S$ on that contrast, removing the intercept, all main effects and all pairs, and test its coefficient with an HC3 standard error. We evaluate TabPFN-3, TabICLv2, TabFM and CatBoost.

**Calibration.** On each table's covariates, we simulate 200 null residuals containing main effects, pairs and heteroskedastic noise, but no term of order three. We apply the same selection and test. A table is counted only if the Wilson 95% upper bound on null rejection is at most 0.10. Eleven of 16 tables qualify: the three CH65 assays, CR9114 H1 and H3, HDV, SGEMM, car, nursery, kr-vs-kp and tic-tac-toe. Their null rejection rates range from 1.5–5.5%. Every model rejects the GAMETES control ($p < 10^{-15}$).

Table 13: **TFMs omit pure third-order components on more calibrated tables than CatBoost.** Rejections at $p < 0.05$, with counts after Holm correction within each model in parentheses, over 11 calibrated tables. *Omission* also requires the selected component to be present in the target and insufficiently captured. Shares measure the fraction of $E$-residual variance explained.

| Model | Rejections | Measured targets | Median / max share | Omission tables |
|---|---|---|---|---|
| TabPFN-3 | 7 (6 Holm) | 5 of 7 | 0.28% / 5.1% | G189E, MA90, CR9114-H3, HDV |
| TabICLv2 | 7 (6 Holm) | 6 of 7 | 0.20% / 1.6% | G189E, MA90, CR9114-H3, HDV |
| TabFM | 7 (6 Holm) | 6 of 7 | 0.12% / 1.8% | G189E, MA90, SI06, CR9114-H3, HDV |
| CatBoost | 3 (2 Holm) | 3 of 7 | 0.04% / 0.7% | MA90, SGEMM |

**The same component for every model.** Different model residuals select different components. To make a direct comparison, Table 14 scores all models on one component per landscape and on the target's projection onto main effects and pairs. TabPFN-3 residuals select the component for G189E and HDV; CatBoost residuals select it for CR9114-H3. Each component is strongly present in its target ($p \leq 4 \times 10^{-6}$). TFMs capture the signal of lower order fully, except for the incorrectly scaled TabPFN-3 base on CR9114-H3 noted in the table, but capture only $-0.07$ to $0.41$ of the component of order three. CatBoost captures $0.78$–$0.93$, leaving no significant remainder. On MA90, SGEMM and car, all models, including CatBoost, miss part of their selected component. The omitted components explain only a median $0.1$–$0.3\%$ of residual variance; TFMs still have better overall accuracy on most of these tables.

Table 14: **Every model scored on the same pure component of order three.** Entries give capture of lower-order / three-way structure on held-out rows. Capture 1 agrees with the target.

| Landscape | Selected by | TabPFN-3 | TabICLv2 | TabFM | CatBoost |
|---|---|---|---|---|---|
| CH65 G189E | TabPFN-3 residual | 0.99 / 0.32 | 0.99 / 0.13 | 0.99 / 0.31 | 0.99 / 0.78 |
| CR9114 H3 | CatBoost residual | 0.67 / 0.22[†] | 0.98 / 0.39 | 0.99 / 0.41 | 0.99 / 0.93 |
| HDV ribozyme | TabPFN-3 residual | 1.24 / 0.17 | 1.21 / 0.05 | 1.22 / $-0.07$ | 1.33 / 0.91 |

[†] TabPFN-3 has incorrectly scaled base predictions (slope 1.49); this cell does not isolate the interaction.

## K Residual search and support restriction

**Algorithm.** We first obtain TFM predictions out of fold using two folds of the training rows, and compute residuals $r$. We transform columns into bounded median-sign and centered-rank channels $Z$, partition coordinates into $L$ balanced blocks $I_\ell$, and draw a random sign vector $v_\ell$ supported on each block. For distinct $i, j$, we compute

$$M_{\ell,ij} = \frac{1}{n}\sum_t r_t Z_{ti} Z_{tj}(Z_t^\top v_\ell) - v_{\ell i}\frac{1}{n}\sum_t r_t Z_{ti}^2 Z_{tj} - v_{\ell j}\frac{1}{n}\sum_t r_t Z_{ti} Z_{tj}^2 = \sum_{k \in I_\ell \setminus \{i,j\}} v_{\ell k}\widehat{T}_{ijk}. \quad (1)$$

Within each block we retain the $B$ largest absolute pair scores, decode the third coordinate, and rank candidates by their direct residual moments. Subtracting repeated indices gives exact sample identities, removing terms through which additive residuals could dominate a contraction. The chosen support trains a residual predictor, which is added to a base fitted on all training rows. Test rows are used in neither step. For a fixed decoding budget, search takes $O(Lnp^2 + Bnp)$ time and $O(np + p^2 + nB)$ memory. It finds the exact support in 150 of 153 aligned parity fits on the benchmark. Figure 5 compares residual refits on all columns, discovered supports and oracle supports using identical residual targets.

**Detector scaling.** We assess scaling on fresh simulations with one interaction triple of amplitude $0.6$, unit noise and 8,000 rows. With 1,024 columns, balanced search recovers 37/40 supports from sign features and 36/40 from continuous uniform features. Dense contractions recover 4/40 and 15/40 with the same proposal budget of eight anchors. Median search times are $0.75$ and $0.85$ seconds, against about 31 seconds for exhaustive scanning. Sampling 20,000 random triples finds no support at this width. At 512 columns with additive contamination, corrected balancing finds 19/40 supports, compared with 6/40 for dense contractions and 0/40 for uncorrected balancing.

### K.1 CONTROLS USING INDEPENDENT SAMPLES

We divide each task into 8,000 rows $A$ for one base fit, 4,000 independent rows $B$ for discovery, 4,000 rows $C$ to fit corrections, 2,000 rows $V$ to choose a weight from $\{0, .25, .5, 1\}$, and 4,000 test rows. Each model and regime has ten fresh tasks. Wide parity uses 512 columns and four squared competitors of weight $0.35$. The nonmonotone regime uses 128 columns, two linear competitors of weight $0.30$, and $g = 0.6 \prod_{j=1}^{3} (2\mathbf{1}[|X_j| > \Phi^{-1}(.75)] - 1)$.

The residual TFM receives all columns (RES), the discovered support, all columns with that support duplicated, or the true support. A product arm fits ridge with penalty 10 on the top discovered product. Oracle comparisons fit CatBoost on the true support, selecting depth and tree count on an 80/20 split of $C$, and ridge on the true formula.

Table 15: **Support and function controls with independent samples.** RMSE over ten tasks per regime. Recall measures exact recovery of the top support. The final three arms use oracle information.

| Model | Rule | RES | Found | Duplicated | Product | True | True $q$ | True $g$ | Recall |
|---|---|---|---|---|---|---|---|---|---|
| TabICLv2 | parity | 1.175 | 1.052 | 1.175 | 1.024 | 1.052 | 1.028 | 1.011 | 10/10 |
| TabPFN-3 | parity | 1.172 | 1.025 | 1.159 | 1.022 | 1.025 | 1.026 | 1.009 | 10/10 |
| TabICLv2 | all-even | 1.167 | 1.167 | 1.167 | 1.167 | 1.153 | 1.054 | .996 | 0/10 |
| TabPFN-3 | all-even | 1.167 | 1.167 | 1.167 | 1.167 | 1.074 | 1.051 | .996 | 0/10 |

Using discovered supports on wide parity improves RMSE over refitting all residual columns by $10.5\%$ for TabICLv2 and $12.5\%$ for TabPFN-3. Matched random supports give almost no improvement. Duplicating the support helps little, whereas the single product is at least as effective as a TFM correction. For the all-even rule, every product of odd transforms has zero population correlation by symmetry, explaining the 0/10 discovery rate. Even on the true support, TabICLv2 underperforms the separate function learner. Thus, finding the support and learning the function remain distinct obstacles.

### K.2 THE KNOWN-RULE TRACK

The known-rule track has fifteen public classification tables with documented label rules: tic-tac-toe, kr-vs-kp, MONK's problems 1–3, the 6-bit multiplexer, parity5+5, xd6, corral and six GAMETES tables. We encode columns as integers, reserve 25% of rows for testing (at most 5,000), and train on the remainder (at most 5,000), using three splits. Each TFM supplies probabilities out of fold with two folds and residuals $y - \hat{p}$. Balanced search over categorical indicators returns four supports. An 80/20 training split selects the base, product augmentation or support restriction.

TFMs are nearly perfect on MONK's, xd6, corral and kr-vs-kp and outperform CatBoost and TabM on multiplexer-6 (Table 16). They fail on parity5+5, whose order five exceeds the dictionary's three columns, and on GAMETES three-way epistasis. For the latter, search finds the epistatic SNPs in every split and lowers log loss 12–20%; CatBoost and TabM also benefit. Search rarely finds the pair in the 1,000-SNP two-way table, but support restriction still lowers log loss 10–16%. The gate harms TabICLv2 on tic-tac-toe ($0.038 \rightarrow 0.049$) and 75:25 heterogeneity ($0.557 \rightarrow 0.604$), and TabFM on parity5+5 ($0.664 \rightarrow 0.699$).

### K.3 CORRECTION WITH A SIGNIFICANCE GUARD

We let the guard choose between a refit on discovered columns, a residual refit on all columns, and least squares on the discovered product. Training rows are divided in half; each half runs the complete pipeline, including search, to predict the other. We accept a policy only if its one-sided paired $t$-test against the base TFM, on differences in squared error per row, has $p < 0.05/3$ and mean squared error improves by at least 1%. We refit the best accepted policy on all training rows, retaining the base if none passes. The native cohort comprises 35 regression families plus `puma32H`, using 4,000 training rows. For the planted cohort we add $0.6\,\mathrm{sign}(z_1 z_2 z_3)$ on three fresh Gaussian columns to the same 35 families.

No TFM has a native table harmed by more than 1% after guarding; the largest increase is $+0.4\%$. Refitting all columns without a guard harms 14–19% of tables (Table 17). The guard corrects 34

Table 16: **Known-rule track.** Mean test log loss over three splits for CatBoost (CB), TabM and each TFM, before (base) and after correction gated by validation. Hit is the fraction of discovered supports containing a documented clause; – indicates that no clause fits in three columns.

| | | | TabPFN-3 | | | TabICLv2 | | | TabFM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Table | CB | TabM | base | gated | hit | base | gated | hit | base | gated | hit |
| tic-tac-toe | 0.020 | 0.092 | 0.045 | 0.023 | 0.58 | 0.038 | 0.049 | 0.42 | 0.027 | 0.031 | 0.42 |
| kr-vs-kp | 0.019 | 0.028 | 0.023 | 0.023 | – | 0.019 | 0.019 | – | 0.015 | 0.014 | – |
| MONK's-1 | 0.002 | 0.054 | 0.000 | 0.000 | 1.00 | 0.000 | 0.000 | 1.00 | 0.000 | 0.000 | 0.42 |
| MONK's-2 | 0.006 | 0.056 | 0.000 | 0.000 | – | 0.000 | 0.000 | – | 0.000 | 0.000 | – |
| MONK's-3 | 0.112 | 0.095 | 0.092 | 0.088 | 0.42 | 0.091 | 0.092 | 0.33 | 0.095 | 0.092 | 0.42 |
| multiplexer-6 | 0.210 | 0.361 | 0.076 | 0.013 | 1.00 | 0.037 | 0.038 | 0.83 | 0.007 | 0.007 | 0.58 |
| parity5+5 | 0.453 | 0.340 | 0.694 | 0.695 | – | 0.696 | 0.695 | – | 0.664 | 0.699 | – |
| xd6 | 0.002 | 0.057 | 0.000 | 0.000 | – | 0.000 | 0.000 | – | 0.000 | 0.000 | – |
| corral | 0.021 | 0.074 | 0.001 | 0.000 | 1.00 | 0.003 | 0.000 | 1.00 | 0.000 | 0.000 | 1.00 |
| GAMETES 2-way, $h^2$=0.1 | 0.639 | 0.629 | 0.613 | 0.613 | 0.00 | 0.617 | 0.617 | 0.00 | 0.614 | 0.614 | 0.00 |
| GAMETES 2-way, $h^2$=0.4 | 0.457 | 0.470 | 0.423 | 0.432 | 0.25 | 0.427 | 0.433 | 0.50 | 0.428 | 0.429 | 0.00 |
| GAMETES 3-way, $h^2$=0.2 | 0.652 | 0.699 | 0.688 | 0.547 | 1.00 | 0.623 | 0.549 | 1.00 | 0.646 | 0.546 | 1.00 |
| GAMETES heterog. 50:50 | 0.569 | 0.565 | 0.507 | 0.513 | 0.00 | 0.538 | 0.538 | 0.75 | 0.503 | 0.505 | 0.08 |
| GAMETES heterog. 75:25 | 0.562 | 0.575 | 0.558 | 0.527 | 1.00 | 0.557 | 0.604 | 1.00 | 0.541 | 0.527 | 1.00 |
| GAMETES 2-way, 1,000 SNPs | 0.690 | 0.828 | 0.694 | 0.608 | 0.33 | 0.789 | 0.708 | 0.00 | 0.727 | 0.608 | 0.17 |

Table 17: **The guard limits observed harm while retaining planted gains.** Geometric mean test RMSE ratio (guarded/base), with 95% intervals from bootstrapping families. Harm is the fraction of tables worsening by more than 1%; gains count tables where the guard applies a correction and lowers RMSE by at least 1%.

| | Native (36 tables) | | | Planted parity (35 tables) | |
|---|---|---|---|---|---|
| Model | Guarded ratio | Harm (guarded / full) | Gains | Guarded ratio | Retained |
| TabPFN-3 | 0.985 [0.966, 0.998] | 0% / 19% | 19% | 0.579 [0.491, 0.680] | 99% |
| TabICLv2 | 0.988 [0.969, 0.999] | 0% / 14% | 14% | 0.579 [0.500, 0.664] | 102% |
| TabFM | 0.991 [0.975, 1.000] | 0% / 17% | 8% | 0.482 [0.387, 0.590] | 101% |

or 35 of 35 planted families and retains 99–102% of the best ungated gain. Exact recovery, with discovered support outperforming random support, occurs in 97% of families. Most native gains come from generic stacking through the full residual refit: `puma32H` ($-25\%$ for every TFM), naval propulsion ($-12\%$), methane ($-7\%$) and Pol ($-11\%$, TabICLv2). Among the 17 discrete tables in Appendix J, no guarded fit worsens by more than 1%, while unguarded refits harm 6–8 tables. With 256 validation rows per context, the guard also removes every harm from the equal label budget corrections in Appendix I, including CERN ($+40$ to $+89\%$ without guarding) and log-runtime SGEMM ($+17$ to $+27\%$). However, FPS improvements are reduced to $-2.6\%$ and $-5.2\%$.

## L UNMODIFIED TARGETS AND NEGATIVE RESULTS

**An interaction in a robot arm simulator.** To test omissions without planting interactions, we searched twelve datasets from seven families of simulators or explicit formulas: Delve pumadyn, kin and bank, F16 control simulations, fried, 2dplanes and mv. Each contributes 8,192 rows, split into 2,500 for discovery, 3,692 for fitting and 2,000 for testing. Discovery uses TabPFN-3 residuals from predictions made out of fold, balanced search, and removal of a fitted additive and pairwise projection. A component advances if it predicts the fitting targets ($p < 0.01$) and TabPFN-3 captures less than half of it. The clearest omission is in `puma32H`. Additive and pairwise learners on its selected columns capture only $0.08$ and $0.01$, compared with $0.95$ for depth-6 CatBoost, showing structure of higher order. TFMs capture $0.08$–$0.09$ at full width. Retaining support $S = \{3, 4, 14\}$ and one column chosen during discovery, $I = \{15\}$, raises capture to $1.00$, $0.88$ and $0.85$ (Figure 13). Jointly permuting the remaining 28 columns changes capture by less than $0.001$, attributing this omission to dilution.

The `pumadyn32nh` component is weaker and unstable; `delta_elevators` fails to replicate on test rows; TabPFN-3 already captures `kin8nm` (0.98). Adding twelve engineered main effects gives crowding of $0.20$, $0.04$ and $-0.21$ for TabPFN-3, TabICLv2 and TabFM.

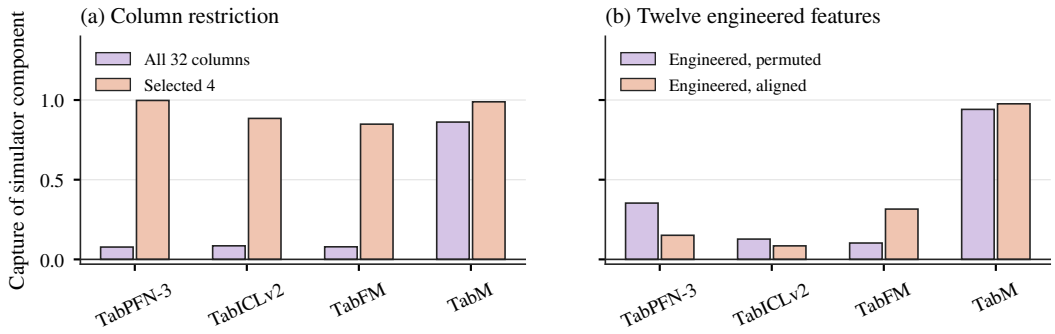


Figure 13: **An unplanted interaction in a robot arm simulator is lost at full width.** Mean capture on `puma32H` over three seeds. (a) Selecting four of 32 columns restores capture. (b) Adding twelve engineered main effects, aligned or permuted. Discovery rows alone determine feature construction and selection.

**A residual test on continuous tables.** We also evaluated a residual test on 35 held-out public regression families, searching on one split and testing on another after removing a fitted projection of lower order. At nominal level 5%, it rejects 12–36% of synthetic nulls containing only structure of lower order. A raw third-moment test rejects 94–99%. With dependent inputs, the fitted projection leaves some of this simpler structure in residuals. We therefore rely on the exact discrete test in Appendix J when reporting measured omissions.

**Repairs on ordinary tables.** Without planted interactions, we find no consistent gain from support restriction, residual column labels or pruning. Relative to correction on matched random supports across the 35 families, geometric mean RMSE ratios are $0.995$ $[0.989, 0.999]$ for TabPFN-3, $0.997$ $[0.989, 1.004]$ for TabICLv2 and $0.996$ $[0.987, 1.004]$ for TabFM. Giving TabPFN-3 its own residuals from predictions made out of fold as column labels on 28 unmodified tables yields ratio $1.011$ $[0.960, 1.055]$, or $0.999$ with validation gating. Importance pruning without a gate increases RMSE 31–36%.

**Feature engineering and fine-tuning.** Spline features computed out of fold appear to harm TFMs on 34 families. The harm vanishes when the feature learner is fitted on held-out rows, indicating a mismatch in stacking. Fine-tuning on 4,000 rows with paired interaction objectives improves planted capture, but the task sampler includes the evaluated families. At 67 columns, no objective recovers parity even in the absence of informative competitors.

# M THEORY

## M.1 WHY PROPER SUBSETS REVEAL NO PARITY SIGNAL

Write $y = g(x_S) + c(x_{-S}) + \varepsilon$.

**Proposition 1** (Proper-subset independence)**.** *Let $g(x_S) = a \prod_{j \in S} \operatorname{sign}(x_j - m_j)$, where $m_j$ is the median and $|S| \geq 2$. If the columns of $S$ are independent of each other and of $(x_{-S}, \varepsilon)$ and symmetric about their medians with no atom there, then for every $j \in S$ and every column set $A$ that omits $j$ and at least one other member of $S$, $x_j$ is independent of $(y, x_A)$.*

*Proof.* Define $\sigma_l = \operatorname{sign}(x_l - m_l)$ and $a_l = |x_l - m_l|$ for $l \in S$. With symmetry and no atom at the median, $\sigma_l$ is a fair sign independent of $a_l$. Choose $u \in S \setminus (A \cup \{j\})$ and condition on every variable except $\sigma_u$. The product in $g$ then has a fair sign for either value of $\sigma_j$. Integrating over the other variables gives independence of $(y, x_A)$ and $\sigma_j$. Also, $a_j$ is independent of the signs, other columns and noise, and affects neither $y$ nor $x_A$. It follows that $(y, x_A)$ is independent of $x_j = m_j + \sigma_j a_j$. □

### M.2 THE ALLOCATION MODEL

Partition the columns as $[p] = S \cup C \cup N$ and assign scores $s_j = D(x_j, u)$. Here $D$ depends on the joint law of $(x_j, u)$, with $D \geq D_0$ and $D = D_0$ exactly when $x_j \perp u$. For a strictly increasing $\phi > 0$, define $w_j = \phi(s_j)/\sum_l \phi(s_l)$ and $G_S = \prod_{j \in S} w_j$. In the permuted arm, competitors are replaced by independent copies with unchanged marginals, independent of the target.

**Proposition 2** (Label-gated allocation)**.** *Let $g$ be a hidden parity on $|S| = k$ columns satisfying Proposition 1. Assume $x_C \perp \varepsilon$, $x_N \perp (x_S, x_C, \varepsilon)$, and any added nuisance $\nu \perp (x, \varepsilon)$. Let $p = |S| + |C| + |N|$, $\phi_0 = \phi(D_0)$ and $\Phi_C = \sum_{j \in C} \phi(s_j)$ in the aligned arm with $u = y$.*

*(i)* Invisibility. *$s_j = D_0$ for every $j \in S$ in both arms, for any $c$ and independent nuisance $\nu$ added to $u$.*

*(ii)* Crowding. *$G_S^{\text{aligned}} = \rho^k \, G_S^{\text{perm}}$ with $\rho = p\phi_0/\{(p - |C|)\phi_0 + \Phi_C\} \leq 1$, with equality iff no competitor column depends on $u$. This is a gate ratio; an exponential capture ratio requires an additional capture–gate relation and comparable $\rho$ across orders.*

*(iii)* Variance is not information. *Adding independent $\nu$ of any variance to $y$ leaves $\rho = 1$ in the permuted arm.*

*(iv)* Residual labels release. *If allocation instead uses $u = y - c(x_C)$, then $\rho = 1$ regardless of the labels used for prediction.*

*(v)* Dilution and restriction. *In the permuted arm $G_S = p^{-k}$, and restricting the input to $S$ attains $G_S = k^{-k}$, the maximum of $\prod_{j \in S} w_j$ over all allocations.*

*Proof.* For (i), apply Proposition 1 to each $j \in S$ and label choice $u \in \{y, y + \nu, y - c(x_C)\}$. It gives $x_j \perp u$, hence $s_j = D_0$. For (ii), every column in the permuted arm scores $D_0$, giving $G_S = p^{-k}$. Support and noise columns retain this score in the aligned arm, where competitors contribute $\Phi_C \geq |C|\phi_0$. Substituting yields $w_j = \rho/p$ on $S$ and the claimed gate ratio. The equality condition follows from strict monotonicity of $\phi$. For (iii), independent nuisance leaves each permuted column independent of $u$. For (iv), the residual label $u = g(x_S) + \varepsilon$ is independent of competitors, restoring their score to $D_0$. Finally, restricting to the support gives weights $1/k$ on $S$. AM–GM bounds any allocation by $\prod_{j \in S} w_j \leq (\sum_{j \in S} w_j/k)^k \leq k^{-k}$, which proves (v). □

A claim about capture additionally requires capture to increase with $G_S$. Our TFM experiments examine the model's qualitative predictions. Encoders acting on groups, conditional tree splits and feature weights learned jointly through gradients can all use information beyond the score of one column.

### M.3 RECOVERY AND RISK STATEMENTS

**Proposition 3** (Balanced recovery)**.** *Let $(Z_t, R_t)$ be iid with independent Rademacher coordinates, $\mathbb{E}R = 0$ and $\|R\|_{\psi_2} \leq \tau$. Suppose the only nonzero third-order moment is on a triple $S_\star$, with value $a$. With $q = \lceil p/L \rceil$, a data-independent balanced partition, $B \geq 3$ and decoding budget $d \geq 1$, search returns $S_\star$ first with probability at least $1 - \delta$ whenever $n \geq C\tau^2 q a^{-2} \log(2Lp^3/\delta)$ for a universal $C$.*

**Proposition 4** (Discovery and validation)**.** *Fix a pointwise base and transforms independently of discovery, correction and validation samples. Suppose search returns a unit-variance sign product $h_\star$, with $a = \mathbb{E}[Rh_\star]$, and conditional residual $R$ is sub-Gaussian. Fit a clipped ridge correction on an independent sample $C$ and use another sample $V$ to choose among $M$ bounded corrections including zero. With the errors $e_C, e_V$ defined in Appendix M.5, with high probability*

$$\mathcal{R}(f_A) - \mathcal{R}(\widehat{f}_V) \geq a^2 - e_C^2 - 2e_V, \qquad \mathcal{R}(\widehat{f}_V) \leq \mathcal{R}(f_A) + 2e_V.$$

### M.4 BALANCED SEARCH: PROOFS AND STATISTICAL SCOPE

#### M.4.1 EXACT CONTRACTION IDENTITY

Expand $Z_t^\top v$ in Equation 1 for an observed matrix $Z$ and residual vector $r$. Terms with $k = i$ or $k = j$ give exactly the two subtractions; all remaining terms contain three distinct coordinates. This

is a sample identity, requiring no population assumption. For sign features, $Z_{ti}^2 = 1$ reduces the correction to $bv^\top + vb^\top$, with $b = n^{-1}Z^\top r$. More generally, define $A_{ij} = n^{-1}\sum_t r_t Z_{ti} Z_{tj}^2$ to obtain $A\,\mathrm{diag}(v) + \mathrm{diag}(v)A^\top$. We rank only entries off the diagonal.

### M.4.2 Proof of Proposition 3

Use the convention $\|W\|_{\psi_2} = \inf\{c > 0 : \mathbb{E}\exp(W^2/c^2) \le 2\}$ and condition on the partition and signs. For a block and a distinct pair, write

$$U_{\ell,ij} = Z_i Z_j \sum_{k \in I_\ell \setminus \{i,j\}} v_{\ell k} Z_k.$$

The sum is a signed Rademacher sum, and multiplying by $Z_i Z_j$ leaves its magnitude unchanged. Thus its sub-Gaussian norm is at most $C_0\sqrt{q}$. The product $RU_{\ell,ij}$ has sub-exponential norm at most $C_1\tau\sqrt{q}$ even though the factors need not be independent. Applying Bernstein's inequality and a union bound to at most $Lp^2$ entries gives

$$\max_{\ell, i<j} \left| \frac{1}{n}\sum_t R_t U_{t,\ell,ij} - \mathbb{E}[RU_{\ell,ij}] \right| \le C_2\tau\sqrt{q}\left(\sqrt{t/n} + t/n\right), \qquad t = \log(C_3 Lp^3/\delta),$$

with probability at least $1 - \delta/3$, after adjusting universal constants. Each term of $U$ multiplies three distinct independent signs, so $\mathbb{E}U = 0$. Centering residuals empirically adds $-\overline{R}\,\overline{U}$. Concentration for these two sub-Gaussian means, followed by a union bound, bounds its magnitude by $C_4\tau\sqrt{q}\,t/n$ with probability at least $1 - \delta/3$.

For direct moments, $|R\prod_{j\in S} Z_j| = |R|$, so the sub-Gaussian norm is at most $\tau$. A union bound across all $\binom{p}{3}$ triples limits deviations to $C_5\tau\sqrt{t/n}$, with centering adding at most $C_6\tau t/n$. Using the whole dictionary also covers proposals and decoding chosen adaptively on the same sample. With sufficiently large $C$, the sample condition in Proposition 3 makes all contraction and direct errors smaller than $|a|/4$. The bound $|a| \le C_7\tau$ absorbs the linear Bernstein term.

A population contraction can be nonzero only for the three pairs in $S_\star$. Each true pair has magnitude $|a|$ in the block containing its third coordinate. Since a block has at most three such pairs, $B \ge 3$ keeps them. For each retained true pair, the correct third coordinate scores at least $3|a|/4$; incorrect choices score at most $|a|/4$. With $d \ge 1$, decoding therefore includes $S_\star$ and the final direct ranking places it first. A union bound over these events finishes the proof. □

For example, $R = a\prod_{j\in S_\star} Z_j + b^\top Z + \varepsilon$ satisfies the moment conditions when noise is independent and centered. Distinct sign products are orthogonal, so the additive term contributes no third moments. Its sub-Gaussian norm is bounded by a universal constant times $|a| + \|b\|_2 + \|\varepsilon\|_{\psi_2}$. The nuisance need not be small, although larger nuisance requires more samples.

**Cost and limits.** The $L$ matrix contractions cost $O(Lnp^2)$. Decoding $B$ pairs per block costs $O(Bn\sum_\ell |I_\ell|) = O(Bnp)$, and ranking at most $LBd$ triples directly costs $O(nLBd)$. Memory is $O(np + p^2 + nB)$. With fixed $L, B, d$, computation is quadratic in the number of features, while the sufficient sample size still depends on $q = \lceil p/L \rceil$. Under the ideal observation model, exhaustive moments avoid that factor; dense anchors correspond to $q = p$. The guarantee thus trades statistical power for computation rather than establishing statistical optimality. It excludes arbitrary multiple interactions, estimated marginal transforms, dependent coordinates and pooled residuals from cross-fitting. The theorem concerns sign inputs; results for continuous inputs are empirical.

**Corollary 1** (Multiple supports without shared pairs)**.** *In Proposition 3, replace the single nonzero third moment by exactly $s$ unordered triples with nonzero moments $a_S$, and define $a_{\min} = \min_S |a_S| > 0$. Keep all other observation assumptions. If no two triples share a coordinate pair, $B \ge 3s$, and $d \ge 1$, the first $s$ entries in the final ranking are exactly these triples with probability at least $1 - \delta$ provided*

$$n \ge C\frac{\tau^2\lceil p/L \rceil}{a_{\min}^2}\log\left(\frac{2Lp^3}{\delta}\right).$$

*Recovering every support also requires retaining at least $s$ final features.*

*Proof.* Apply the uniform bounds for contractions and direct moments from Proposition 3, taking error below $a_{\min}/4$. A pair occurs in at most one true triple. Its contraction therefore has either zero population mean or that triple's single signed coefficient, so there is no cancellation within a pair. No block has more than $3s$ entries with nonzero means, and the proposal budget keeps each true pair in the block of its third coordinate. That pair has one nonzero direct third moment, which decoding finds. All $s$ true moments have magnitude at least $3a_{\min}/4$, and all false moments at most $a_{\min}/4$. The final ranking follows. □

Unequal amplitudes enter through $a_{\min}^{-2}$. The implementation retains only four features, so cannot recover every support when $s > 4$. More interactions can also raise $\tau$, and the larger budget $B$ increases decoding cost; the guarantee depends on $s$. Sharing a pair permits cancellation within a contraction and is excluded here.

### M.4.3 Residual projection and information source

Let $H$ be a vector of square-integrable features, with $G = \mathbb{E}[HH^\top]$ and $b = \mathbb{E}[HR]$. Expanding squared loss gives improvement $2w^\top b - w^\top Gw$. For $v \in \ker G$, we have $v^\top H = 0$ almost surely and consequently $v^\top b = 0$, placing $b$ in the range of $G$. The maximum is $b^\top G^\dagger b$, achieved by $w = G^\dagger b$ plus any vector in the null space: the usual least squares projection. If $H$ is orthogonal to a specified additive function space, residual components in that space do not affect $b$. Independent inputs and centered products ensure this orthogonality; centering alone does not suffice with arbitrary dependence.

Now fix $f$ and $P_X$. A detector seeing only these objects returns identically distributed outputs under $Y = f(X) + \varepsilon$ and $Y = f(X) + ah(X) + \varepsilon$. Any binary randomized decision has type-I and type-II errors summing to one on this pair. Attribution of a fixed predictor therefore cannot certify that it represents the entire target.

## M.5 From discovery to population-risk improvement

We prove Proposition 4 conditional on the base sample $A$, all transforms fitted on $A$, and randomness chosen without $B, C, V$. Under this conditioning, $f_A$ is a deterministic pointwise function and $R = Y - f_A(X)$ is assumed sub-Gaussian. The result excludes transductive predictors whose output for one query depends on other queries.

**Formal statement.** Take independent iid samples $B, C, V$, a sign product $h_\star \in \{-1, 1\}$ with $\mathbb{E}h_\star^2 = 1$, $\|R\|_{\psi_2} \le \tau$, and $a = \mathbb{E}[Rh_\star]$. Suppose the detector finds $h_\star$ on $B$ outside an event of probability $\delta_D$. On $C$, fit ridge using summed loss, $\widehat{a} = \sum_{i \in C} R_i h_\star(X_i)/(n_C + \lambda)$, and clip to $\widetilde{a} = \mathrm{clip}(\widehat{a}, [-B_0, B_0])$, where $B_0 \ge |a|$. On $V$, minimize uncapped squared loss over at most $M$ predictors, including the base and $f_A + \widetilde{a}h_\star$, with corrections bounded by $B_0$. For a universal $C > 0$, Proposition 4 then holds with probability at least $1 - \delta_D - \delta_C - \delta_V$, where

$$e_C = \frac{\lambda|a|}{n_C + \lambda} + \frac{n_C}{n_C + \lambda} C\tau \sqrt{\frac{\log(2/\delta_C)}{n_C}},$$

$$e_V = C(B_0^2 + B_0\tau) \left\{ \sqrt{\frac{\log(2M/\delta_V)}{n_V}} + \frac{\log(2M/\delta_V)}{n_V} \right\}.$$

**Correction estimation.** Set $Z_i = R_i h_\star(X_i)$. From $|h_\star| = 1$ and $\|R\|_{\psi_2} \le \tau$, we obtain $\|Z_i\|_{\psi_2} \le \tau$ and $\mathbb{E}Z_i = a$. Sub-Gaussian concentration bounds, with probability at least $1 - \delta_C$,

$$|\overline{Z}_C - a| \le C\tau \sqrt{\log(2/\delta_C)/n_C}.$$

Substituting $\widehat{a} = n_C \overline{Z}_C/(n_C + \lambda)$ gives

$$|\widehat{a} - a| \le \frac{\lambda|a|}{n_C + \lambda} + \frac{n_C}{n_C + \lambda} C\tau \sqrt{\log(2/\delta_C)/n_C} = e_C.$$

Clipping to $[-B_0, B_0]$ cannot increase distance from $a$, which lies in the interval. Therefore $|\widetilde{a}-a| \leq e_C$. Expanding squared loss with $\mathbb{E}h_\star^2 = 1$ gives the identity

$$\mathcal{R}(f_A + wh_\star) = \mathcal{R}(f_A) - a^2 + (w-a)^2.$$

On the correction event, the discovered candidate consequently has risk at most $\mathcal{R}(f_A) - a^2 + e_C^2$.

**Validation selection.** Compare a fixed correction $q_j$, bounded by $|q_j| \leq B_0$, with the shared base using the paired loss difference

$$D_j = (R - q_j)^2 - R^2 = q_j^2 - 2Rq_j.$$

The shared $R^2$ term cancels. Because $R$ is sub-Gaussian and $q_j$ bounded, $D_j - \mathbb{E}D_j$ is sub-exponential with $\psi_1$ norm at most $C(B_0^2 + B_0\tau)$. Bernstein's inequality, union bounded across $M$ candidates, gives

$$\max_{j \leq M} |(P_V - P)D_j| \leq e_V$$

with probability at least $1 - \delta_V$. The empirical risk minimizer thus has population risk no greater than the best candidate's risk plus $2e_V$. Comparing it with the discovered correction and with the included base gives the two inequalities of Proposition 4.

**Combining events and boundaries.** Condition on $B$ and the discovered function for the coefficient bound. For validation, condition on $B$, $C$, every discovered function, each fitted coefficient and earlier randomness. This fixes the candidate list before observing the independent sample $V$. The events need not be independent: a union bound on the conditional probabilities gives total failure at most $\delta_D + \delta_C + \delta_V$. A positive guaranteed improvement requires $a^2 > e_C^2 + 2e_V$. Independent splits are essential here; the argument does not cover the pooled out-of-fold procedure used experimentally.

## N CODE AND REPRODUCTION

We provide CrowdBench, experiment code, protocols, fit records and analysis outputs in the supplementary material. The package includes the probe requiring two fits, residual search and the known-rule track. After installing dependencies as described in the bundle's README, `python script/review.py --primary` recomputes primary benchmark metrics from saved predictions without fitting models. Every recomputed primary metric agrees within $6 \times 10^{-9}$. Secondary studies include code and records, but not all raw data or full prediction banks.