A SIMPLE RECIPE FOR DISTILLING TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tabular foundation models (TFMs) perform strong zero-shot predictions through in-context learning, but their dependence on labeled context makes repeated inference costly. Knowledge distillation offers a way to transfer this predictive strength to lightweight students. However, the context-dependent nature of TFM inference makes the distillation process non-trivial, raising two essential design questions about what constitutes the context and what constitutes the query. We study these questions across two TFMs and both neural and tree-based students, finding a simple and effective recipe for TFM distillation. On TabArena, our distilled students outperform supervised tuned-and-ensembled students by 57–98 Elo points. Moreover, the same distillation recipe improves students on 236–258 of 300 TALENT datasets and reduces median primary error by 4.0–6.4%. Finally, distilled students achieve median inference speedups of 3.0–21.6× over their teachers, making inference substantially cheaper. Code is available at link.

1 INTRODUCTION

Tabular machine learning has been dominated by supervised methods such as tree models (Chen & Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018) and multi-layer perceptrons (Holzmüller et al., 2024; Gorishniy et al., 2025; 2026). These methods support fast inference once trained but require dataset-specific fitting and often extensive hyperparameter tuning. Recently, tabular foundation models (TFMs) (Hollmann et al., 2025; Ma et al., 2025; Qu et al., 2025; 2026) have changed this paradigm with strong out-of-the-box performance without per-dataset updates. The central idea of TFMs is *in-context learning,* which enables zero-shot inference by conditioning predictions on labeled samples in the new dataset, not on the training samples observed earlier.

However, the in-context learning mechanism of TFMs does not eliminate the computational costs but rather shifts them from training to inference because the context must be processed every time a prediction is made. Repeated TFM inference is therefore much more costly than using a trained supervised method. Several recent works reduce this burden through context selection (Thomas et al., 2024; Liu & Ye, 2025), routing queries to specialized local contexts (Xu et al., 2025), compressing contexts into latent representations (Zabërgja et al., 2026), or caching reusable context representations (Grinsztajn et al., 2025; Qu et al., 2026). These approaches improve the overall efficiency of inference but retain context-dependent computation at serving time.

Knowledge distillation (KD) (Buciluǎ et al., 2006; Hinton et al., 2015) gives a complementary approach that transfers the predictive ability of a TFM to a lightweight, dataset-specific student. Once created, a distilled student supports instant inference as many times as we want, saving huge computational costs. However, because of the unique properties of TFMs as in-context learners, distilling a TFM introduces two key essential design choices: *(i) supervision construction* asks how teacher supervision should be formed from training data, while *(ii) query coverage* asks whether and how expanding teacher queries improves distillation.

Supervision construction. What we distill from a TFM to a student is not its full capacity; we distill its predictive power conditioned on a specific labeled context. For every labeled observation in the training data, we can either (i) include it in the context to make the TFM's predictions more focused around the data manifold, or (ii) exclude it from the context but include it in the query data, possibly providing its true label as an additional supervised signal. We find that students perform best when

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.

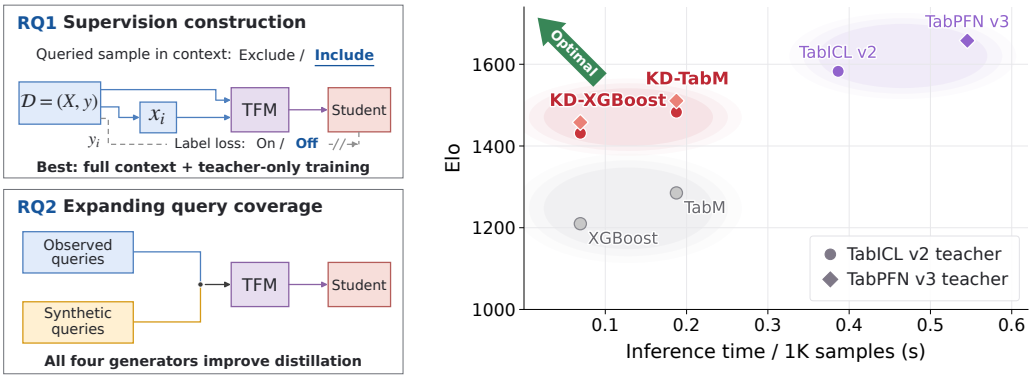

Figure 1: **Constructing supervision for TFM distillation.** (Left) We study how to construct teacher supervision for ditillation (RQ1) and whether and how expanding teacher queries improves distillation (RQ2). We find that training solely on teacher predictions generated with the full labeled training set as context achieves the best performance. (Right) Elo versus median inference time per 1K samples on TabArena for default lightweight students, their distilled counterparts, and the TFM teachers, with Elo calibrated jointly over the same 76 configurations as in Section 4.

trained solely on teacher predictions generated using the full labeled training set as context. Under this design, TFM supervision can fully replace direct label supervision.

Query coverage. The second choice concerns which inputs are used to obtain teacher supervision. One way to make full use of the teacher is to expand query coverage beyond the observed training rows. By querying synthetic inputs, we can obtain additional supervision for the student even without collecting new labels. We test whether this expansion improves distillation using four simple, task-agnostic query generators that perturb or recombine observed features. All four generators improve aggregate performance for every teacher–student pair. These results show that useful teacher supervision can extend beyond observed rows even with simple query generation.

Figure 1 summarizes these design choices and illustrates the resulting trade-off between predictive performance and inference cost on TabArena.

Together, these findings yield a simple but effective distillation recipe of TFMs: use the full labeled training set as teacher context and train the student solely on teacher predictions for observed and synthetic queries. Our distilled students with single default student configurations outperform tuned-and-ensembled students on TabArena (Erickson et al., 2025) by 57–98 Elo points. Moreover, the same recipe improves students across binary classification, multiclass classification, and regression on 300 TALENT datasets (Liu et al., 2025). The distilled students trade a one-time distillation cost and some accuracy relative to their teachers for faster repeated inference, achieving median speedups of 3.0–21.6×, with the inference advantage growing as the teacher context size increases.

2 Related Work

Tabular foundation models. Prior-data fitted networkss (PFNs) learn to approximate posterior predictions by pretraining over synthetic tasks and conditioning on labeled samples at inference time (Müller et al., 2022). TabPFN introduced this paradigm to tabular prediction (Hollmann et al., 2023; 2025), and recent TFMs have substantially improved predictive performance and scalability (Qu et al., 2025; 2026; Grinsztajn et al., 2025; 2026; Cho et al., 2026). Unlike supervised models, which fit a separate predictor to each dataset, they adapt through in-context learning: predictions are created by conditioning on labeled examples from the target dataset without dataset-specific parameter updates. This enables strong out-of-the-box prediction, but it also means that inference continues to depend on the labeled context even after the model is fully deployed.

Improving context efficiency. The context dependence of TFMs has motivated several approaches to improve efficiency. One direction improves the efficiency of the context itself. Lo-

CalPFN constructs local contexts through nearest-neighbor retrieval (Thomas et al., 2024), while MixturePFN routes queries to specialized experts with cluster-specific contexts (Xu et al., 2025). BETA combines lightweight encoder adaptation with bootstrapped sampling to scale contextual prediction (Liu & Ye, 2025). TACO jointly trains a context compressor and a predictor, enabling predictions to condition on a compact latent representation of the training data (Zabërgja et al., 2026). TabPFN-2.5 and TabICLv2 accelerate repeated inference by caching reusable context representations (Grinsztajn et al., 2025; Qu et al., 2026). These approaches reduce inference cost, but predictions still depend on context examples or their cached representations at serving time.

Knowledge distillation of TFMs. A complementary direction transfers the information of a TFM into a dataset-specific predictor that can be served independently. MotherNet generates the weights of a compact neural network from a dataset (Müller et al., 2025), while knowledge distillation trains a lightweight student from teacher predictions. Classical distillation may combine teacher predictions with ground-truth labels (Buciluǎ et al., 2006; Hinton et al., 2015). For tabular models, FAST-DAD distills ensemble teachers into individual predictors using observed labels together with teacher predictions on synthetic inputs generated by Gibbs sampling. (Fakoor et al., 2020). Recent TFM distillation methods likewise instantiate particular choices of teacher supervision: Pocket Foundation Models combines out-of-fold teacher predictions with direct label supervision (Tanna et al., 2026). In concurrent work, GEAR combines distillation on synthetic samples with sophiscated generators and that combines direct label supervision and out-of-fold teacher supervision (Qin et al., 2026).

3 Designing TFM Distillation

We study two design choices for distilling a pretrained TFM into a lightweight, dataset-specific student. The first, *supervision construction*, concerns how teacher predictions are generated for training datas and whether they are combined with direct label supervision. The second, *query coverage*, concerns whether and how querying additional inputs improves distillation.

Experimental setup. Our design studies use 51 TabArena datasets with three official outer splits. We consider all four teacher–student pairs formed by two strong TFM teachers, TabICLv2 (Qu et al., 2026) and TabPFN v3 (Grinsztajn et al., 2026), and two structurally distinct students, the neural model TabM (Gorishniy et al., 2025) and the tree-based model XGBoost (Chen & Guestrin, 2016). Our experiments build on the official TabArena training and evaluation pipeline, retaining its default student configurations, preprocessing, and protocol. Configurations are compared using aggregate TabArena Elo, calibrated jointly within each comparison. Further details appear in Appendix A.1.

3.1 Supervision Construction

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ denote the labeled training dataset. A TFM prediction $q_T(\cdot \mid x, C)$ depends on a labeled context C and a query input x. The context provides evidence for prediction, while the output for the query serves as supervision for the student. When querying a training data, we can either exclude it from the context or include it together with its label.

We compare two protocols that generate teacher targets for the same samples. Out-of-fold (OOF) prediction queries each held-out fold using the remaining folds as context, excluding the queried sample and its label. Full-context (FC) prediction queries each training sample using the full labeled training set $\mathcal{D}$ as context, including the queried sample and its label. Let $\{\mathcal{D}_k\}_{k=1}^K$ partition $\mathcal{D}$, and let $f(i)$ denote the fold containing (x_i, y_i):

$$q_i^{\mathrm{OOF}} = q_T\big(\cdot \mid x_i, \mathcal{D} \setminus \mathcal{D}_{f(i)}\big), \qquad q_i^{\mathrm{FC}} = q_T(\cdot \mid x_i, \mathcal{D}). \tag{1}$$

Either set of teacher targets can be further combined with direct label supervision through the objective

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[\alpha\, \ell_{\mathrm{KD}}(q_{S_\theta}(\cdot \mid x_i), q_i) + (1-\alpha)\, \ell_{\mathrm{sup}}(q_{S_\theta}(\cdot \mid x_i), y_i)\right], \tag{2}$$

where q_{S_θ} denotes the student prediction and q_i is either q_i^{OOF} or q_i^{FC}. The losses ℓ_{KD} and ℓ_{sup} measure agreement with the teacher target and the ground-truth label, respectively. Thus, $\alpha = 1$

uses only teacher predictions, $\alpha = 0$ uses only ground-truth labels, and intermediate values combine the two sources of supervision.

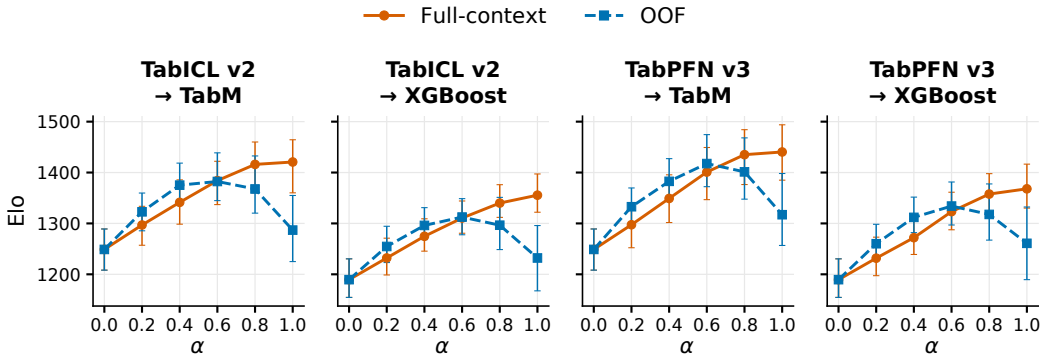

Figure 2: **Teacher-only supervision with the full labeled context achieves the best aggregate performance.** Full-context targets reach the highest Elo at $\alpha = 1$ in every teacher–student pair, exceeding the best OOF–label mixture at $\alpha = 0.6$.

Full-context teacher predictions provide the best supervision. Across all four teacher–student pairs, the best tested configuration trains the student solely on teacher predictions generated with the full labeled training set as context. We sweep $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ for both protocols. Figure 2 shows that full-context targets achieve their highest aggregate Elo at $\alpha = 1$, with no direct label loss. In contrast, OOF targets benefit from adding direct label supervision, with Elo peaking at $\alpha = 0.6$ in every pair. Full-context teacher-only supervision exceeds the best tested OOF–label mixture by 22.8–43.0 Elo points.

Finding 1. Teacher predictions generated with the full labeled training set as context can replace direct label supervision. For both teachers and student families studied, learning solely from these predictions yields the best aggregate performance among the tested supervision configurations.

We therefore fix the full labeled training set as teacher context and use teacher-only supervision ($\alpha = 1$) for the remaining experiments.

3.2 Expanding Query Coverage

One simple way to make fuller use of the teacher is to expand query coverage beyond the real training rows. By querying synthetic inputs, we can obtain additional supervision without collecting new labels. We ask whether this expansion improves distillation when the queries are generated by simple perturbations or recombinations of real sample's features. For each synthetic input $\tilde{x}$, we obtain a teacher target using

$$q^{\text{syn}} = q_T(\cdot \mid \tilde{x}, \mathcal{D}), \tag{3}$$

where the labeled context $\mathcal{D}$ remains the same as for real queries. The student then learns from teacher predictions on both real and synthetic queries.

Four simple query generators. We evaluate four task-agnostic, non-parametric generators at a common budget of one synthetic query per real row. CutMix independently selects each feature from one of two sampled rows with equal probability. Mixup interpolates numerical features between two rows using feature-wise mixing coefficients, while categorical features select between the corresponding values. FullMix independently draws each feature from a randomly selected real row. Perturb adds standard-deviation-scaled Gaussian noise to numerical features and resamples categorical values from their empirical marginals.

Expanding query coverage improves distillation. Figure 3 compares distillation using only real queries with adding synthetic queries from each generator. All four yield higher Elo for every teacher–student pair, with gains ranging from 14 to 91 Elo points. CutMix achieves the highest Elo in all four pairs, and each of the other generators also improves over real queries alone.

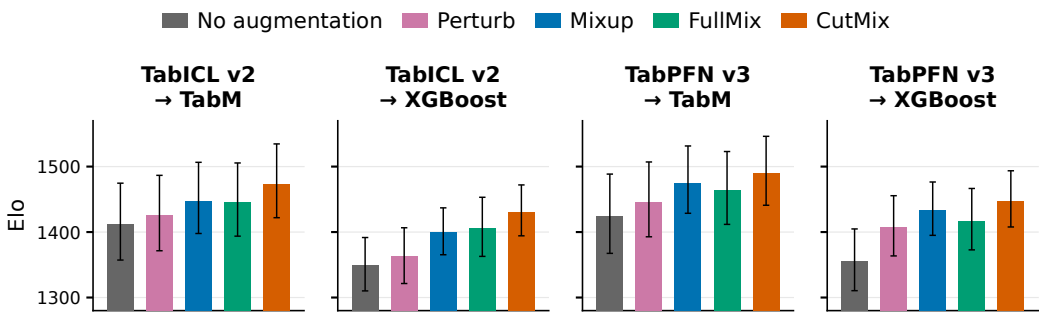

Figure 3: **Simple synthetic queries improve distillation across all four teacher–student pairs.** Each of the four query generators yields higher Elo than distillation using only observed queries, with CutMix achieving the highest Elo in every pair.

Finding 2. Simple synthetic queries extend useful teacher supervision beyond real training rows. The gains hold across all four tested generators, both teachers, and both students.

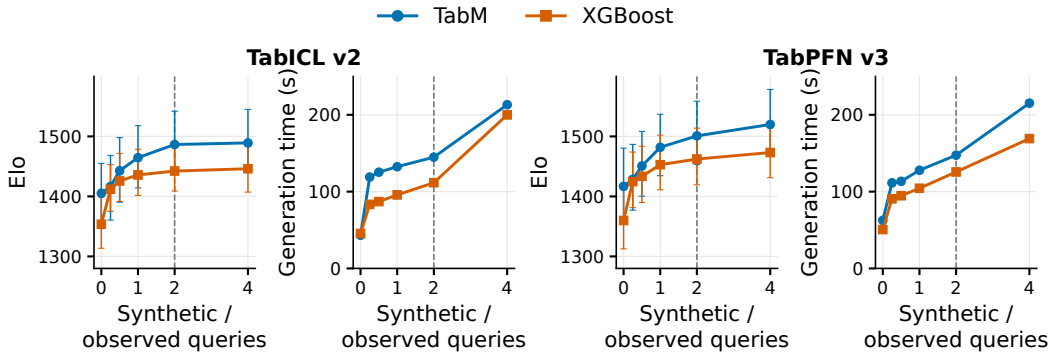

Figure 4: **Effect of the synthetic-query budget for CutMix.** Each teacher group pairs mean Elo with teacher prediction generation time. Increasing the synthetic-to-real query ratio improves mean Elo across both teachers and student families while increasing generation time. The dashed line marks the 2:1 ratio used in our fixed recipe, chosen as a common operating point on this performance–generation-cost trade-off rather than tuned per dataset.

More synthetic queries provide further gains at additional teacher cost. Using CutMix, Figure 4 jointly tracks Elo and teacher prediction generation time as the synthetic-to-real query ratio increases. Increasing the ratio from 1:1 to 2:1 adds 6.5–21.9 Elo across the four pairs, while mean generation time increases by 9–20%. The benefit of a larger budget varies across pairs: moving from 2:1 to 4:1 adds only 2.7 and 3.8 Elo for TabICLv2–TabM and TabICLv2–XGBoost, respectively, at 47% and 79% higher mean generation cost. TabPFN v3–TabM continues to gain 19.0 Elo over the same increase, with 46% more generation time.

Although 4:1 achieves the highest Elo point estimate in every pair, 2:1 already improves substantially over real queries alone while requiring less target-generation time. We therefore use 2:1 as a common performance–cost compromise for the remaining experiments.

Final recipe. Together, these studies define a single distillation recipe that we fix for all subsequent experiments. We use the full labeled training set as teacher context and train the student solely on teacher predictions and CutMix synthetic queries at a 2:1 synthetic-to-real ratio.

4 Evaluation

Section 3 fixes a single recipe for TFM distillation. We now evaluate this recipe on TabArena (Erickson et al., 2025) and TALENT (Liu et al., 2025).

Table 1: **KD improves each student beyond supervised tuning and ensembling.** Each KD model is compared with the tuned-and-ensembled version of the same architecture on 51 TabArena datasets. KD gain is the median relative primary-error reduction, W/T/L counts dataset-level wins, ties, and losses. Elo is calibrated jointly over all 76 configurations; differences are computed before rounding. Brackets show paired dataset-bootstrap 95% intervals.

Model	ΔElo ↑	KD gain (%) ↑	W/T/L
TabICL v2 → TabM	+56.7 [3.9, 107.4]	1.30 [0.61, 1.93]	33/0/18
TabICL v2 → XGBoost	+71.4 [14.3, 128.7]	1.69 [0.38, 2.21]	35/0/16
TabPFN v3 → TabM	+84.5 [34.1, 135.1]	1.38 [0.71, 2.36]	36/0/15
TabPFN v3 → XGBoost	+97.9 [41.9, 156.9]	1.91 [0.94, 3.15]	37/0/14

Experimental setup. As in Section 3, we use TabICLv2 (Qu et al., 2026) and TabPFN v3 (Grinsztajn et al., 2026) as teachers, and TabM (Gorishniy et al., 2025) and XGBoost (Chen & Guestrin, 2016) as students. For KD, we'll use the selected recipe in Section 3 without further reselection and use the default student configurations throughout. On TabArena (Erickson et al., 2025), we evaluate on all 51 datasets under the official benchmark protocol. We compute aggregate TabArena scores jointly over the 72 official baseline configurations and our four KD configurations, using default student configurations and a 2:1 synthetic-to-observed query ratio ($r = 2$). On TALENT (Liu et al., 2025), we evaluate on all 300 datasets using the provided train/validation/test splits and five training seeds. We measure error by $1 - \mathrm{ROC\text{-}AUC}$ for binary classification, log loss for multiclass classification, and RMSE for regression. For each dataset, we first average errors over splits or seeds and then compute matched relative-error reductions. Aggregate results weight datasets equally, with uncertainty estimated by paired dataset-bootstrap 95% confidence intervals. Appendix A.2 provides full protocols, statistical details, and additional analyses.

Baselines. On both benchmarks, we compare KD against architecture-matched default students trained directly without distillation. On TabArena, we additionally compare against tuned and tuned-and-ensembled versions of the same students, all 15 conventional supervised model families in the benchmark pool, including RealMLP (Holzmüller et al., 2024), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), as well as the teacher models.

4.1 Distillation Improves the Same Student Beyond Direct Optimization

KD outperforms tuned-and-ensembled direct training for both student architectures. Across the four teacher–student pairs, distilled students gain 56.7–97.9 Elo over the matched tuned ensemble, with all paired 95% confidence intervals above zero. They also win on 33–37 of 51 datasets, with median relative-error reductions of 1.30–1.91% and positive confidence intervals for every pair.

The distilled students also rank strongly against the broader TabArena pool. All four outperform the strongest configuration of 14 of the 15 conventional supervised model families. RealMLP is the only exception, although TabPFN v3–TabM also exceeds its tuned and ensembled variant. Table 2 reports selected baselines; the full comparison is in Appendix B.

4.2 Teacher Advantage Transfers through KD

To assess whether KD actually transfers the teacher's predictive advantage, we measure how much of the teacher–student error gap is recovered by the distilled student and whether larger teacher advantages translate into larger KD gains. Across datasets where the teacher reduces raw-student error by more than 1%, KD recovers a median 67.9–70.9% of the teacher–student error gap (Table 3). Thus, when the TFM is meaningfully stronger than the student, roughly two-thirds of its predictive advantage is retained in the lightweight student.

This transfer is also reflected at the dataset level. Larger teacher advantages are associated with larger KD gains (Figure 5), with Spearman correlations of 0.67–0.82 across the four teacher–student pairs and all bootstrap 95% confidence intervals above zero. This relationship indicates that the benefit of KD closely tracks the dataset-specific advantage provided by the teacher. Consistent with

Table 2: **Distilled students and leading supervised models on TabArena.** Results cover 51 datasets and 816 dataset–split pairs using a joint pool of 72 official baseline configurations and four KD configurations. Win rate averages pairwise outcomes against the other 75 methods in this pool. Distilled students use default student configurations and $r = 2$, with no student hyperparameter search. Light-blue row shading identifies distilled students. Among the displayed non-teacher models, **1st**, **2nd**, and **3rd** best results in each column are highlighted. Teachers are reported separately as references and excluded from these markings.

Model	Elo ↑	Improvability (%) ↓	Win rate (%) ↑
TabPFN v3 → TabM	**1510.7**	**11.21**	**76.4**
RealMLP (Tuned+Ensemble)	**1489.9**	**11.09**	**74.5**
TabICL v2 → TabM	**1483.0**	**11.44**	**73.9**
TabPFN v3 → XGBoost	1457.8	12.41	71.4
TabICL v2 → XGBoost	1431.3	12.62	68.8
TabM (Tuned+Ensemble)	1426.2	12.50	68.3
LightGBM (Tuned+Ensemble)	1412.5	13.49	66.8
CatBoost (Tuned+Ensemble)	1398.1	13.10	65.3
TabM (Tuned)	1373.7	13.36	62.7
XGBoost (Tuned+Ensemble)	1359.9	14.26	61.2
XGBoost (Tuned)	1338.1	14.57	58.7
TabM (Default)	1285.2	15.57	52.7
XGBoost (Default)	1209.8	17.35	44.2
Teacher models			
TabPFN v3	1657.8	6.83	87.2
TabICL v2	1582.7	7.60	82.2

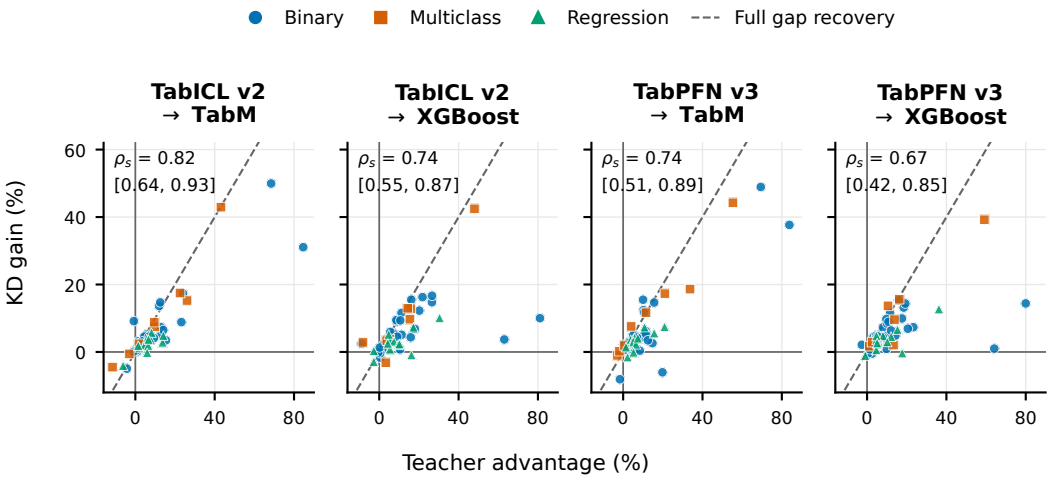

Figure 5: **Transfer of teacher gains to distilled students across 51 TabArena datasets.** Teacher and KD gains are measured as relative error reduction over the raw student, $100(1 - E/E_{\text{raw}})$; the dashed line denotes full recovery of the teacher gain. Annotations report Spearman correlations with 95% bootstrap CIs. Because both axes share the raw baseline, these correlations are descriptive.

this pattern, when the teacher outperforms the raw student, KD also improves the student on nearly all such datasets, with only one to three exceptions per pair.

4.3 One-Time Distillation Cost vs. Inference Efficiency

We finally examine whether the one-time cost of distillation can be amortized through faster repeated inference, and in which serving regimes this trade-off is most favorable. Table 4 quantifies the

Table 3: **Recovery of teacher gains on TabArena.** We restrict to datasets where the teacher reduces default-student test error by more than 1%. Gap recovery is $100(E_{\text{raw}} - E_{\text{KD}})/(E_{\text{raw}} - E_T)$, using errors averaged over official splits; entries report the dataset median with bootstrap 95% CIs, and W/T/L against the raw student.

Model	n	Gap recovery (%)	KD W/T/L
TabICL v2 → TabM	43	70.5 [57.4, 90.0]	42/0/1
TabICL v2 → XGBoost	43	69.4 [54.0, 80.6]	41/0/2
TabPFN v3 → TabM	46	70.9 [55.3, 91.6]	43/0/3
TabPFN v3 → XGBoost	48	67.9 [52.7, 78.1]	46/0/2

Table 4: **Inference savings, prediction error, and distillation cost.** Results on 51 TabArena datasets at outer split 0. Latencies are per 32-row request; speedup uses dataset-paired teacher/KD ratios, and ΔError reports the relative error increase with paired-bootstrap 95% CIs. Offline time indicates end-to-end distillation tuime. Point estimates are dataset medians and brackets denote IQRs.

Model	Teacher (ms)	KD (ms)	Speedup	ΔError (%)	Offline (s)
TabICL v2 → TabM	232.2	47.8	3.0×	1.72 [0.53, 2.94]	186.3 [93.1, 450.9]
TabICL v2 → XGBoost	232.2	29.1	8.7×	2.33 [1.07, 5.54]	48.0 [20.0, 146.5]
TabPFN v3 → TabM	566.9	46.7	11.1×	2.76 [0.70, 4.12]	192.4 [110.4, 563.4]
TabPFN v3 → XGBoost	566.9	29.7	21.6×	2.53 [1.33, 4.17]	59.6 [30.6, 173.9]

deployment trade-off across all 51 TabArena datasets using 32-row requests on the same H200 host with 32 pinned CPU cores. Serving measurements start from resident models and exclude initial loading, whereas offline cost includes teacher-target generation, student fitting, and associated overhead (Appendix A.3). Across the four teacher–student pairs, dataset-paired median inference speedups range from 3.0 to $21.6\times$, while the distilled students incur median relative primary-error increases of 1.72–2.76% and median one-time costs of 48.0–192.4 seconds.

To relate this upfront cost to repeated serving, let A denote the one-time distillation cost and t_T and t_S the teacher and student latencies per 32-row request, measured in the same time units. We estimate the number of requests required to amortize distillation as

$$N_{\text{break-even}} = \frac{A}{t_T - t_S}. \tag{4}$$

If $t_T \leq t_S$, no finite break-even point exists and we set $N_{\text{break-even}} = \infty$. Table 5 groups datasets by teacher context size and reports medians of the dataset-level speedups and break-even request counts within each group.

The deployment benefit grows markedly with teacher context size. With at least 10K context rows, the distilled student is faster on all 15 datasets for every teacher–student pair, with median speedups of 23.9–$53.7\times$, compared with 1.9–$13.1\times$ below 1K rows. For TabICLv2–TabM, the larger serving advantage also reduces the median break-even volume from about $3{,}261$ to 468 requests. However, speedup and amortization are not equivalent: the break-even point also depends on the upfront distillation cost and the absolute latency saved per request. For example, for TabPFN v3–XGBoost, the median speedup rises from $13.1\times$ to $32.3\times$, while the median break-even volume remains near 100 requests. Overall, distillation is most compelling when the teacher incurs substantial context-dependent inference cost and the resulting student is reused for repeated prediction.

4.4 Evaluation on TALENT

We use TALENT as an external validation benchmark to test whether the fixed distillation recipe could be generalized. We apply the recipe unchanged to all 300 TALENT datasets. Across all four

Table 5: **Inference savings and break-even requests by teacher context size.** Entries report dataset medians of inference speedup t_T/t_S and break-even 32-row requests. All 51 datasets are included; students are faster on all 15 datasets with at least 10K teacher context rows.

Context rows	$<$ 1K ($n = 10$)		1K–$<$ 10K ($n = 26$)		$\geq$ 10K ($n = 15$)	
Model	Speedup	$N_{\text{break-even}}$	Speedup	$N_{\text{break-even}}$	Speedup	$N_{\text{break-even}}$
TabICL v2 → TabM	1.9×	3,260.5	2.1×	2,073.6	31.4×	468.3
TabICL v2 → XGBoost	3.0×	393.0	6.8×	211.2	23.9×	160.1
TabPFN v3 → TabM	10.2×	503.2	6.8×	388.9	53.7×	266.5
TabPFN v3 → XGBoost	13.1×	97.5	20.3×	97.6	32.3×	100.8

Table 6: **The fixed recipe improves both students across TALENT task types.** KD gains are median relative primary-error reductions against the same default supervised student, after averaging five matched seeds. Brackets show paired dataset-bootstrap 95% intervals. W/T/L counts wins, ties, and losses on all 300 datasets; gain summaries exclude three zero-error baselines.

Model	Median KD gain (%) ↑				W/T/L
	All $n = 297$	Binary $n = 117$	Multiclass $n = 80$	Regression $n = 100$	$n = 300$
TabICL v2 → TabM	**4.20** [3.14, 5.33]	8.46 [5.55, 10.41]	5.80 [2.81, 10.65]	1.44 [0.47, 2.99]	238/3/59
TabICL v2 → XGBoost	**5.69** [4.60, 7.50]	9.53 [7.32, 12.23]	9.08 [7.21, 13.46]	2.71 [1.74, 3.49]	253/2/45
TabPFN v3 → TabM	**4.02** [3.13, 6.15]	7.90 [5.00, 9.98]	7.63 [3.36, 10.09]	1.66 [0.75, 3.13]	236/3/61
TabPFN v3 → XGBoost	**6.37** [5.33, 7.48]	9.82 [7.38, 12.94]	9.65 [6.61, 14.43]	3.49 [2.37, 4.91]	258/2/40

teacher–student pairs, KD improves over supevised trained students on 236–258 datasets, reducing median primary error by 4.02–6.37%, with all paired 95% confidence intervals above zero.

The improvement is not concentrated in a particular task. Across binary classification, multiclass classification, and regression, all 12 teacher–student–task combinations show positive median improvements with 95% confidence intervals above zero. Together, these results show that the fixed recipe transfers across a substantially broader collection of datasets without adapting new supervision design.

5 Conclusion

We studied *supervision construction* and *query coverage* as two key design choices for distilling TFMs into lightweight, dataset-specific students. Across all four teacher–student pairs, training solely on full-context teacher predictions achieves the best aggregate performance among the tested supervision configurations. All four simple, task-agnostic query generators further improve aggregate performance for every pair, showing that useful teacher supervision can extend beyond observed training rows.

These findings yield a fixed recipe that trains students solely on full-context teacher predictions for observed and synthetic queries. With default student configurations, this recipe outperforms architecture-matched tuned-and-ensembled students on TabArena and improves matched default students on TALENT without reselecting its target-generation settings. Predictive gains vary across datasets and are associated with the teacher's advantage over the student trained directly on observed labels. The distilled students can be served independently of the labeled context, trading a one-time distillation cost and some accuracy relative to their teachers for faster repeated inference, with the largest speedups in large-context settings.

AI USE STATEMENT

Generative AI tools were used to support code development and manuscript preparation, mainly by providing coding suggestions and assisting with improvements to clarity and grammar. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted text was independently examined and revised. Generative AI was not used in producing experimental results or in making research-related decisions. The authors assume full responsibility for the final content of the manuscript, including any material developed with AI assistance.

ETHICS STATEMENT

This work studies efficient tabular prediction through knowledge distillation using existing benchmark datasets. Although reducing inference costs can make predictive models more accessible, distilled students may inherit biases and errors from their teachers and training data. Our evaluation focuses on predictive performance and computational efficiency and does not establish fairness, privacy, or suitability for high-stakes applications. In particular, serving a student without access to the teacher's labeled context does not itself provide a privacy guarantee. Appli

REPRODUCIBILITY STATEMENT

Our source code is available via the link provided in the abstract. Section 3 describes the distillation formulation and synthetic-query construction, while Section 4 specifies the benchmarks, baselines, and evaluation metrics. Appendices A.1 and A.2 document the implementation and evaluation protocols, including the benchmark dependency version, data splits, teacher-context construction, training settings, random seeds for TALENT, and statistical procedures. Appendix A.3 specifies the hardware and timing protocol used to measure inference latency and offline costs. Full TabArena comparison results are provided in Appendix B.

REFERENCES

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 535–541, 2006.

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.

Minyong Cho, Minho Jeong, Dooho Lee, Jinmo Lee, and Jaemin Yoo. Causilo technical report. *arXiv preprint arXiv:2609.22866*, 2026. URL `https://arxiv.org/abs/2609.22866`.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. In *Neural Information Processing Systems (NeurIPS)*, 2025.

Rasool Fakoor, Jonas W Mueller, Nick Erickson, Pratik Chaudhari, and Alexander J Smola. Fast, accurate, and simple models for tabular data via augmented distillation. *Advances in Neural Information Processing Systems*, 33:8671–8681, 2020.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. Tabm: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations*, volume 2025, pp. 77899–77935, 2025.

Yury Gorishniy, Akim Kotelnikov, Ivan Rubachev, and Artem Babenko. TabPack: Efficient Hyperparameter Ensembles for Tabular Deep Learning. In *ICML*, 2026.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, et al. Tabpfn-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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.

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*, 01 2025. doi: 10.1038/s41586-024-08328-6.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: strong pre-tuned mlps and boosted trees on tabular data. In *Neural Information Processing Systems (NeurIPS)*, 2024.

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. *Advances in neural information processing systems*, 30, 2017.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Huai-Hong Yin, Tao Zhou, Jun-Peng Jiang, and Han-Jia Ye. TALENT: A tabular analytics and learning toolbox. *Journal of Machine Learning Research*, 26(226):1–16, 2025. URL https://www.jmlr.org/papers/v26/25-0512.html.

Siyang Liu and Han-Jia Ye. TabPFN unleashed: A scalable and effective solution to tabular classification problems. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5DD3RCcVcT.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=pIZxEOZCId.

Andreas Müller, Carlo Curino, and Raghu Ramakrishnan. Mothernet: Fast training and inference via hyper-network transformers. In *International Conference on Learning Representations*, volume 2025, pp. 76666–76686, 2025.

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

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Qi Qin, Jiajie Zhu, Dali Chen, Yuzhao Zhang, Jia-Xing Han, Yu Su, Peng Zhang, Ying Yan, and Yifan Sun. GEAR: Generative expansion and real anchoring for two-stage distillation of tabular foundation models. *arXiv preprint arXiv:2608.18849*, 2026.

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.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *International Conference on Machine Learning*, 2026.

Aditya Tanna, Nassim Bouarour, Mohamed Bouadi, Vinay Kumar Sankarapu, and Pratinav Seth. Pocket foundation models: Distilling TFMs into cpu-ready gradient-boosted trees. *arXiv preprint arXiv:2605.18654*, 2026.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval & fine-tuning for in-context tabular models. *Advances in Neural Information Processing Systems*, 37:108439–108467, 2024.

Derek Qiang Xu, F Olcay Cirit, Reza Asadi, Yizhou Sun, and Wei Wang. Mixture of in-context prompters for tabular PFNs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2fojNANZSv.

Guri Zabërgja, Rafiq Kamel, Arlind Kadra, Christian M. M. Frey, and Josif Grabocka. End-to-end compression for tabular foundation models. In *International Conference on Machine Learning*, 2026.

A Experimental Details

A.1 Section 3 details

Training protocol. We use TabArena pinned to commit `eb5bc081` (16 June 2026), and its 72-method official comparison preset. Section 3 use TabArena splits 0–2 on 51 datasets with eight-fold student bagging. Full-context targets use all fitting rows of each bagging member as teacher context; validation and test labels are excluded. OOF targets use up to eight inner folds within that fitting set.

Implementation. We retain benchmark-default students and preprocessing, use eight teacher estimators, and apply no additional distillation temperature scaling. Classification uses soft cross-entropy and regression uses squared error to teacher predictions. TabM and XGBoost balance observed and synthetic queries through the sum of their group-mean losses.

Synthetic queries. The operator comparison uses one synthetic query per observed row. CutMix replaces each feature with probability 0.5; Mixup uses independent feature-wise $\mathrm{Unif}(0, 1)$ coefficients. Perturb uses numerical noise scaled by the column standard deviation and categorical replacement probability 0.5; FullMix samples each feature independently from observed rows. All source rows are sampled from the fitting set, and synthetic targets come only from the teacher. The TabPFN v3–XGBoost budget comparison uses 152 matched dataset–split pairs because one $r = 4$ result is unavailable; the other curves use 153.

A.2 Section 4 details

Evaluation protocol. TabArena evaluation covers all 816 official dataset–split pairs: nine splits for 34 datasets and 30 for 17, including design splits 0–2. Appendix C repeats the analyses in Sections 4.1 and 4.2 after excluding design splits 0–2. TALENT uses its fixed train/validation/test partitions and seeds 0–4, with training-only teacher context and no outer bagging or seed ensembling. The released TALENT adapter uses the same `eb5bc081` dependency with a TabM compatibility layer. The official tuned TabArena baselines select from one default and 200 random configurations; tuned ensembles use 40 ensemble-selection iterations.

Statistics. We average errors over matched splits or seeds before computing dataset-level relative gains, then report their median. Paired bootstrap intervals resample datasets, using 20,000 replicates with joint recalibration of all 76 configurations for Section 4 Elo contrasts and 10,000 for TALENT median gains. TALENT relative gains exclude reference errors at most 10^{-12}, leaving 297 datasets; win/tie/loss counts use all 300, with absolute differences at most 10^{-12} treated as ties. Each dataset receives equal weight.

A.3 Timing details

Serving measurements. We measure all 51 TabArena datasets at outer split 0 on one H200 host with 32 pinned CPU cores. The reported latencies use resident models, 32-row requests, and teacher context recomputation without persistent KV caching. After three warm-up calls, we record 7–30 GPU-synchronized calls, targeting two seconds of measured time, and take the median latency per dataset. Initial model loading is excluded.

Cost aggregation. Offline cost includes teacher-target generation, student fitting, and training overhead. Speedups and break-even request counts are computed per dataset before taking medians, retaining infinite break-even values when the student is no faster. In Figure 4, generation time is the mean recorded cold target-generation cost across completed splits, excluding student fitting.

B Full TabArena Results

Table 7 reports all 72 official baseline configurations and our four KD students using default student configurations and $r = 2$. Elo, improvability, and win rate are computed jointly over exactly these 76 configurations.

Table 7: Full TabArena results on 51 datasets and 816 dataset–split pairs. All metrics are computed jointly over the 76 displayed configurations: 72 official baselines and four KD students using default student configurations and $r = 2$. Shaded rows denote KD students.

Model	Elo ↑	Improvability (%) ↓	Win rate (%) ↑
AutoGluon 1.5 (extreme, 4h)	1668.8	5.57	87.9
TabPFN v3 (Default)	1657.8	6.83	87.2
TabPFN v2.6 (Default)	1605.4	8.62	83.9
RealTabPFN v2.5 (Tuned+Ensemble)	1585.5	8.26	82.4
TabICL v2 (Default)	1582.7	7.60	82.2
RealTabPFN v2.5 (Tuned)	1543.2	9.02	79.2
RealTabPFN v2.5 (Default)	1514.3	9.53	76.7
KD: TabPFN v3 → TabM ($r = 2$)	1510.7	11.21	76.4
AutoGluon 1.4 (best, 4h)	1490.7	10.99	74.6
RealMLP (Tuned+Ensemble)	1489.9	11.09	74.5
KD: TabICL v2 → TabM ($r = 2$)	1483.0	11.44	73.9
KD: TabPFN v3 → XGBoost ($r = 2$)	1457.8	12.41	71.4
TabDPT (Tuned+Ensemble)	1443.2	11.58	70.0
KD: TabICL v2 → XGBoost ($r = 2$)	1431.3	12.62	68.8
TabM (Tuned+Ensemble)	1426.2	12.50	68.3
LightGBM (Tuned+Ensemble)	1412.5	13.49	66.8
RealMLP (Tuned)	1412.2	12.47	66.8
CatBoost (Tuned+Ensemble)	1398.1	13.10	65.3
TabDPT (Tuned)	1391.8	12.81	64.6
iLTM (Tuned+Ensemble)	1391.7	13.56	64.6
CatBoost (Tuned)	1388.7	13.30	64.3
ModernNCA (Tuned+Ensemble)	1375.3	13.78	62.8
TabM (Tuned)	1373.7	13.36	62.7
LightGBM (Tuned)	1370.2	14.16	62.3
XGBoost (Tuned+Ensemble)	1359.9	14.26	61.2
CatBoost (Default)	1353.2	14.06	60.4
LimiX (Default)	1350.8	13.57	60.2
ModernNCA (Tuned)	1344.2	14.30	59.4
xRFM (Tuned+Ensemble)	1339.7	14.67	58.9
XGBoost (Tuned)	1338.1	14.57	58.7
TabPFN v2 (Tuned+Ensemble)	1331.5	14.92	58.0
Mitra (Default)	1318.7	15.43	56.6
TabDPT (Default)	1318.0	15.19	56.5
TabICL (Default)	1309.3	15.06	55.5
xRFM (Tuned)	1294.7	15.89	53.8
iLTM (Tuned)	1288.4	15.47	53.1
TabM (Default)	1285.2	15.57	52.7
TorchMLP (Tuned+Ensemble)	1275.8	15.48	51.7
TabPFN v2 (Tuned)	1275.2	16.46	51.6
SAP-RPT-OSS (Default)	1273.2	16.60	51.4
BetaTabPFN (Default)	1270.8	17.09	51.1
EBM (Tuned+Ensemble)	1256.5	17.28	49.5
TabPFN v2 (Default)	1248.1	17.23	48.5
ModernNCA (Default)	1242.5	17.61	47.9
RealMLP (Default)	1223.1	16.97	45.7
EBM (Tuned)	1222.9	17.94	45.7
XGBoost (Default)	1209.8	17.35	44.2
TorchMLP (Tuned)	1205.2	17.25	43.7
ExtraTrees (Tuned+Ensemble)	1199.0	18.61	43.0
FastaiMLP (Tuned+Ensemble)	1196.5	18.10	42.7
EBM (Default)	1192.5	18.96	42.3
LightGBM (Default)	1183.1	17.91	41.2

Continued on next page.

Table 7 continued.

Model	Elo ↑	Improvability (%) ↓	Win rate (%) ↑
RandomForest (Tuned+Ensemble)	1174.5	19.54	40.3
ExtraTrees (Tuned)	1165.7	19.53	39.3
FastaiMLP (Tuned)	1138.7	19.55	36.4
RandomForest (Tuned)	1137.3	20.28	36.3
PerpetualBooster (Tuned+Ensemble)	1089.2	25.05	31.3
OrionMSP (Default)	1088.7	22.02	31.3
iLTM (Default)	1088.2	21.76	31.2
TabSTAR (Tuned)	1088.1	24.47	31.2
TabSTAR (Tuned+Ensemble)	1087.2	24.49	31.1
TorchMLP (Default)	1073.8	21.39	29.8
PerpetualBooster (Tuned)	1051.6	26.44	27.7
xRFM (Default)	1041.5	25.05	26.8
TabFlex (Default)	1010.8	26.42	24.1
FastaiMLP (Default)	1004.7	24.27	23.5
RandomForest (Default)	1000.0	25.08	23.1
KNN (Tuned+Ensemble)	992.4	26.76	22.5
TabSTAR (Default)	988.4	29.50	22.2
ExtraTrees (Default)	982.1	26.47	21.7
Linear (Tuned+Ensemble)	956.6	32.71	19.7
Linear (Tuned)	931.9	33.27	17.9
PerpetualBooster (Default)	930.7	30.57	17.8
KNN (Tuned)	886.0	31.87	14.8
Linear (Default)	856.1	35.92	13.0
KNN (Default)	643.7	45.28	4.4

C SECTION 4 RESULTS EXCLUDING DESIGN SPLITS

We repeat the TabArena analyses in Sections 4.1 and 4.2 after excluding outer splits 0–2, which were used for the design studies in Section 3. This leaves 663 dataset–split pairs across all 51 datasets: splits 3–8 for 34 datasets and splits 3–29 for 17 datasets. We retain the fixed full-context, teacher-only CutMix recipe with default student configurations and $r = 2$, without reselecting any design choice. This analysis checks sensitivity to reusing the design splits; the datasets themselves remain the same, so it does not constitute evaluation on unseen datasets.

Aggregation and uncertainty. Elo, improvability, and win rate are recomputed jointly over the same 72 official baseline configurations and four KD configurations, with equal dataset weights. For relative-error analyses, we first average errors over the retained splits within each dataset, then compute dataset-level ratios and report their medians. All intervals below use 20,000 paired dataset-bootstrap resamples and the 2.5th and 97.5th percentiles; Elo intervals jointly refit all 76 configurations in each resample. Win/tie/loss counts use dataset-mean errors and an absolute tie tolerance of 10^{-12}.

C.1 COMPARISON WITH SUPERVISED STUDENTS

The conclusion of Section 4.1 holds after excluding the design splits. All four KD configurations outperform the matched tuned-and-ensembled student, with gains of 56.4–101.6 Elo and median relative-error reductions of 1.37–1.84% (Table 8). Every paired interval is above zero, and KD wins on 33–36 of the 51 datasets.

Table 8: **KD versus tuned-and-ensembled students after excluding design splits.** Results use 51 datasets and 663 dataset–split pairs, omitting splits 0–2. KD gain is the median relative primary-error reduction against the matched tuned ensemble; W/T/L counts dataset-level wins, ties, and losses. Elo is jointly recalibrated over all 76 configurations. Brackets show paired dataset-bootstrap 95% intervals (20,000 resamples).

Model	ΔElo ↑	KD gain (%) ↑	W/T/L
TabICL v2 → TabM	+56.4 [4.2, 106.6]	1.37 [0.45, 2.03]	34/0/17
TabICL v2 → XGBoost	+72.1 [14.3, 130.3]	1.64 [0.50, 2.34]	33/0/18
TabPFN v3 → TabM	+86.0 [34.0, 138.5]	1.55 [0.71, 2.28]	35/0/16
TabPFN v3 → XGBoost	+101.6 [44.3, 162.0]	1.84 [0.87, 3.78]	36/0/15

Table 9 repeats the broader comparison from Table 2 on the retained splits. Each KD configuration exceeds the highest Elo among the configurations of at least 14 of the 15 conventional supervised families. TabPFN v3–TabM exceeds all 15; RealMLP remains the only family whose strongest configuration ranks above the other three KD models.

Table 9: **Broader TabArena comparison after excluding design splits.** The displayed models match Table 2; all metrics use the 76-configuration pool on the remaining 663 dataset–split pairs. KD students use default configurations and $r = 2$. Shading marks KD students; **1st**, **2nd**, and **3rd** results among displayed non-teacher models are highlighted.

Model	Elo ↑	Improvability (%) ↓	Win rate (%) ↑
TabPFN v3 → TabM	**1513.5**	**11.18**	**76.5**
RealMLP (Tuned+Ensemble)	**1490.8**	**11.11**	**74.5**
TabICL v2 → TabM	**1484.0**	**11.43**	**73.9**
TabPFN v3 → XGBoost	1460.6	12.36	71.6
TabICL v2 → XGBoost	1431.1	12.63	68.6
TabM (Tuned+Ensemble)	1427.6	12.49	68.3
LightGBM (Tuned+Ensemble)	1412.5	13.48	66.7
CatBoost (Tuned+Ensemble)	1401.2	13.02	65.5
TabM (Tuned)	1373.9	13.39	62.6
XGBoost (Tuned+Ensemble)	1359.0	14.27	60.9
XGBoost (Tuned)	1337.5	14.59	58.5
TabM (Default)	1284.9	15.65	52.5
XGBoost (Default)	1208.5	17.41	43.9
Teacher models			
TabPFN v3	1659.2	6.85	87.3
TabICL v2	1581.9	7.53	82.1

C.2 Teacher Advantage and KD Gain

The relationship between teacher advantage and KD gain in Section 4.2 also persists. Across all 51 datasets, Spearman correlations range from 0.69 to 0.82, with all bootstrap intervals above zero (Figure 6). The teacher and KD gains share the raw-student denominator, so these correlations remain descriptive.

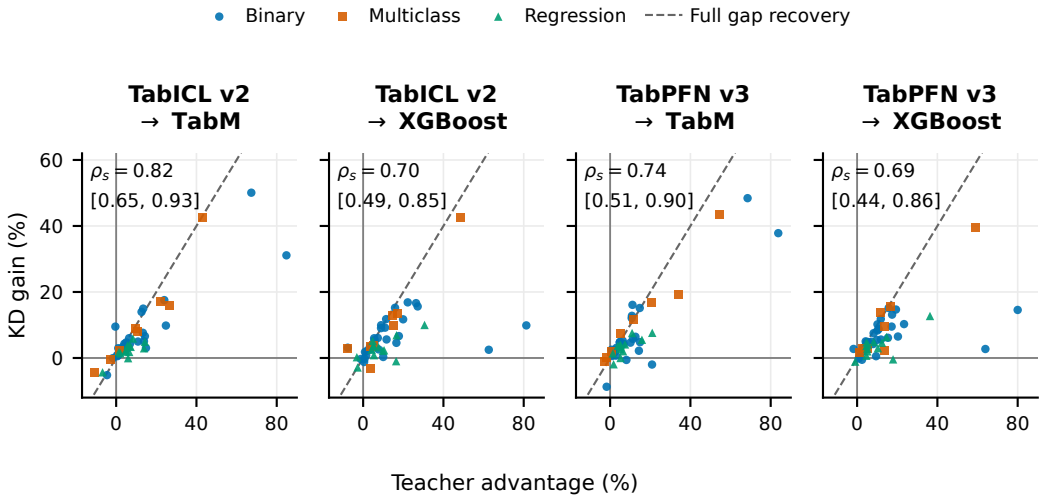

Figure 6: **Teacher advantage and KD gain after excluding design splits.** Each point is one of the 51 TabArena datasets, using errors averaged over splits ≥ 3. Teacher and KD gains are relative error reductions over the default student, $100(1 - E/E_{\text{raw}})$; the dashed line denotes full recovery. Annotations report Spearman correlations with paired dataset-bootstrap 95% intervals (20,000 resamples). These correlations are descriptive because both axes share the raw baseline.

On the 42–48 datasets per pair where the teacher reduces raw-student error by more than 1%, KD recovers a median 67.8–73.4% of the teacher–student error gap (Table 10). The eligible datasets are

selected again using only the retained splits, and recovery intervals resample these eligible datasets. KD still loses to the raw student on one to three eligible datasets per pair, so a stronger teacher does not ensure improvement on every dataset.

Table 10: **Recovery of teacher gains after excluding design splits.** Errors are averaged over retained splits before selecting datasets where the teacher reduces default-student error by more than 1%. Gap recovery is $100(E_{\mathrm{raw}} - E_{\mathrm{KD}})/(E_{\mathrm{raw}} - E_T)$. Entries report the median with dataset-bootstrap 95% intervals (20,000 resamples); W/T/L compares KD with the raw student on these n datasets.

Model	n	Gap recovery (%)	KD W/T/L
TabICL v2 $\rightarrow$ TabM	42	71.8 [56.0, 89.8]	41/0/1
TabICL v2 $\rightarrow$ XGBoost	43	69.0 [53.9, 83.1]	41/0/2
TabPFN v3 $\rightarrow$ TabM	45	73.4 [56.4, 86.2]	42/0/3
TabPFN v3 $\rightarrow$ XGBoost	48	67.8 [52.7, 78.8]	46/0/2