GEAR: GENERATIVE EXPANSION AND REAL ANCHORING FOR TWO-STAGE DISTILLATION OF TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tabular foundation models (TFMs) achieve strong performance through in-context learning, but context-dependent inference imposes substantial latency and memory costs, hindering large-scale deployment. We propose GEAR (*Generative Expansion and Real Anchoring*), a modular two-stage framework that distills TFMs into lightweight MLP or tree-based predictors that can be deployed on commodity CPUs. Stage 1 uses synthetic covariates solely as teacher-query locations and trains the student on soft TFM targets, expanding coverage beyond observed rows. Stage 2 re-anchors the student to the target distribution using real labels and out-of-fold teacher predictions, which avoids self-labeling leakage. Our derived risk certificate characterizes the trade-off between generated-query volume and generator fidelity. Experiments on TALENT and TabArena demonstrate the broad applicability of GEAR. Two-stage MLPs outperform supervised MLPs by 1.81–2.00 AUC points on binary tasks and 1.19–1.35 points on multiclass tasks, with additional gains over real-data-only distillation of 1.76–2.19 and 2.09–2.40 points, respectively. On binary tasks, the gains also transfer to LightGBM and XGBoost, and all three student families outperform CatBoost, the strongest non-TFM baseline, in mean AUC. Ablations show gains beyond longer training or alternative warm starts, greater stability from staged than mixed optimization, and generator-dependent diminishing returns as query volume increases. Finally, GEAR reduces median inference time by 57–2866$\times$ and peak prediction memory by 1.9–3.3$\times$, while retaining higher AUC than matched supervised baselines.

1 INTRODUCTION

Tabular data are central to machine learning in finance, marketing, healthcare, and other high-stakes domains (Grinsztajn et al., 2022). Unlike natural language and images, tables are mixed-type, heterogeneous, and dataset-specific, making reusable representations difficult to learn and deploy (Jiang et al., 2026a). Practical tabular learning has therefore relied on task-specific preprocessing and model selection; gradient-boosted decision trees such as XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018) remain highly competitive with more complex neural architectures (Van Breugel & Van Der Schaar, 2024). Recent tabular foundation models (TFMs), including TabPFN (Hollmann et al., 2022; 2025), TabICL (Qu et al., 2025; 2026), and TabDPT (Ma et al., 2024b), challenge this paradigm through transformer-based in-context learning (ICL). Given a labeled table as context, they predict query rows without task-specific gradient updates. By learning across large collections of synthetic tabular tasks, these models achieve strong few-shot performance.

This in-context interface, however, creates a deployment bottleneck (Tanna et al., 2026b; Zabërgja et al., 2026). During inference, each query batch requires executing a large teacher while conditioning on the labeled context table, so latency and memory consumption can grow sharply with the context size and query workload. Large-context inference is therefore expensive, prone to out-of-memory failures, and requires training records to remain available at prediction time, raising both efficiency and privacy concerns.

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.

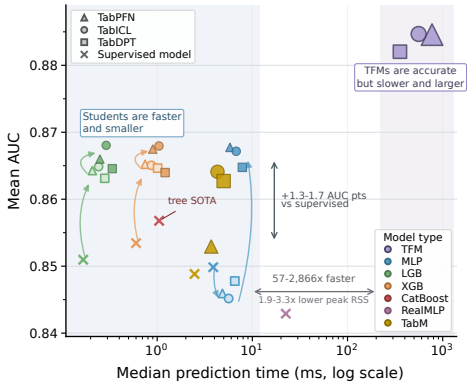


Figure 1: Accuracy–efficiency trade-off on TALENT: mean AUC versus median prediction time, with bubble size indicating peak RSS. Light/dark fills denote real-only/GEAR distillation; crosses mark supervised baselines.

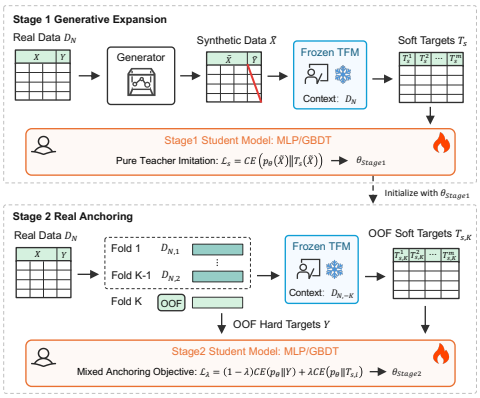


Figure 2: Overview of the proposed two-stage generative distillation protocol.

Recent work mitigates this cost by modifying the context used at inference time, including retrieval or nearest-neighbor context selection (Thomas et al., 2024; Liu & Ye, 2025), mixture-of-experts routing (Xu et al., 2025), divide-and-conquer inference (Ye et al., 2026), and optimal transport-based distillation (Lin & Li, 2026). Although these approaches improve scalability, the predictor still depends on an inference-time context and a large TFM. Consequently, they still incur substantial memory overhead and do not provide the fixed, millisecond-scale latency of a compact, context-free model.

Knowledge distillation (KD) offers a complementary route (Hinton et al., 2015; Tanna et al., 2026b): the teacher's task-adaptive behavior can be amortized into a lightweight MLP or tree-based student. Once distilled, the student has fixed per-row inference cost, runs on commodity CPUs, and no longer requires the labeled context at test time. The main challenge is *query coverage*: the real table provides only finitely many teacher-query locations, which is especially restrictive in small-sample regimes. Moreover, teacher predictions for rows included in the teacher's own context may suffer from self-labeling leakage, motivating out-of-fold (OOF) real-data anchoring.

To address these challenges, we introduce GEAR, a two-stage protocol that performs **G**enerative **ExpA**nsion and **R**eal Anchoring for in-context TFMs, as summarized in Figure 2. Stage 1 queries a frozen teacher on generated covariates and trains the student by teacher-only imitation. Stage 2 initializes from this generated-imitation student and returns to the real table. It uses OOF teacher predictions to avoid self-labeling leakage and optimizes a mixed real-row objective that combines ground-truth labels with teacher soft targets. Thus, synthetic queries expand teacher coverage, whereas real rows provide target-domain anchoring. GEAR is modular and agnostic to generator families, teacher models, and student architectures. It can combine diverse tabular generators and TFM teachers with deployable student families, including MLPs and tree models.

We evaluate GEAR on TALENT and TabArena with three TFM teachers, four generators, varying generated-to-real ratios K, and MLP or tree-based students. GEAR MLPs outperform supervised MLPs by 1.81–2.00 AUC points on binary tasks and 1.19–1.35 points on multiclass tasks. On binary tasks, Stage 2 improves both tree students for every teacher, and all three student families surpass CatBoost, the strongest non-TFM baseline, in mean AUC. Compared with direct TFM inference, GEAR MLPs reduce median prediction time by 57–2866× and peak memory by 1.9–3.3×, while outperforming matched supervised baselines (Figure 1). Volume studies, mechanism ablations, and the risk certificate support separating query expansion from real-data anchoring.

Our contributions are summarized as follows: **(1)** We introduce GEAR, a modular two-stage framework that combines generative query expansion with real-data anchoring to transfer in-context TFMs into efficient predictors while reducing self-labeling leakage and supporting diverse generator families, teacher models, and student architectures. **(2)** Our extensive experiments show that GEAR

improves over supervised learning and real-data-only distillation, while substantially reducing TFM inference costs and preserving competitive predictive performance. **(3)** Our theoretical analysis links student risk to generated-query volume and fidelity. Our analysis explains the diminishing returns from synthetic expansion, the effects of distribution mismatch, and the role of real-data anchoring in the two-stage design.

2 METHODOLOGY

2.1 SETUP

Let P be the joint distribution of (X, Y) on $\mathcal{X} \times [L]$ with covariate marginal P_X, where X is a covariate vector and $Y \in [L] = \{1, \ldots, L\}$ is the class label. We observe a training dataset $D_n = \{(X_i, Y_i)\}_{i=1}^n$ drawn independently from P. Let $\Delta_{L-1} = \{p \in [0,1]^L : \sum_{a=1}^L p_a = 1\}$ denote the probability simplex.

In standard supervised learning, a probabilistic classifier $p_\theta : \mathcal{X} \to \Delta_{L-1}$, parameterized by $\theta \in \Theta$, maps each covariate vector to a predictive distribution. The training data are used only to estimate θ; after fitting, the predictor evaluates new queries independently of the training table. For a supervised loss $\ell : \Delta_{L-1} \times [L] \to \mathbb{R}_+$, such as cross-entropy, the population risk is $R(\theta) = \mathrm{E}_P\{\ell(p_\theta(X), Y)\}$, and empirical risk minimization estimates θ by minimizing its empirical counterpart.

Tabular foundation models (Hollmann et al., 2025) use a different, context-dependent interface. They are pretrained across many synthetic tabular tasks and avoid task-specific parameter updates at deployment. Given a labeled context C and a query x, the teacher returns $f_{\mathrm{ICL}}(x \mid C) \in \Delta_{L-1}$. Thus, unlike $p_\theta(x)$, whose predictions are fixed once training is complete, $f_{\mathrm{ICL}}(x \mid C)$ remains coupled to the size, composition, and labels of the context table.

This context dependence creates a practical deployment burden: each prediction invokes the large teacher and requires access to the labeled table. To amortize this cost while preserving the teacher's task-adaptive behavior, we distill the frozen teacher $f_T(x \mid C) = f_{\mathrm{ICL}}(x \mid C)$ into a lightweight student $p_\theta(x) = f_S(x; \theta)$. The student can be instantiated as an MLP, a tree model, or another deployable tabular model.

2.2 REAL-DATA MIXED DISTILLATION

To distill the TFM teacher's behavior into a lightweight parametric student, we use the standard teacher–student formulation with soft teacher targets (Hinton et al., 2015), adapted to tabular TFM distillation (Tanna et al., 2026b). For a student distribution p and a teacher distribution q in Δ_{L-1}, the soft-label loss is $\ell_T(p, q) = -\sum_{a=1}^L q_a \log p_a$[1]. With mixing weight $\lambda \in [0, 1]$, the real-row mixed loss is

$$\ell_\lambda(p, y, q) = (1 - \lambda)\ell(p, y) + \lambda \ell_T(p, q).$$

With a single fixed teacher context C, the population loss would be $\mathrm{E}_P\, \ell_\lambda\{p_\theta(X), Y, f_T(X \mid C)\}$. On real training rows, however, asking the teacher to predict a row that is also inside its context can leak that row's label into the distillation target. We therefore use out-of-fold (OOF) teacher predictions, following the real-data TFM distillation baseline of Tanna et al. (2026b).

Specifically, we partition D_n into K_{cf} equal folds $I_1, \ldots, I_{K_{\mathrm{cf}}}$ of size $m = n/K_{\mathrm{cf}}$. For fold I_k, let $D_n^{(-k)} = D_n \setminus \{(X_i, Y_i) : i \in I_k\}$ and define the fold teacher map $T_k(\cdot) = f_T(\cdot \mid D_n^{(-k)})$. The fold-averaged OOF population loss is

$$L_\lambda^P(\theta) = \frac{1}{K_{\mathrm{cf}}} \sum_{k=1}^{K_{\mathrm{cf}}} \mathrm{E}_{(X,Y)\sim P}\, \ell_\lambda\{p_\theta(X), Y, T_k(X)\}. \tag{1}$$

The matching empirical OOF objective is

$$\widehat{L}_{\lambda,n}(\theta) = \frac{1}{K_{\mathrm{cf}}} \sum_{k=1}^{K_{\mathrm{cf}}} \frac{1}{m} \sum_{i \in I_k} \ell_\lambda\{p_\theta(X_i), Y_i, T_k(X_i)\}. \tag{2}$$

[1] This is equivalent to minimizing $\mathrm{KL}(q \| p)$ up to the teacher entropy.

The OOF construction prevents self-labeling by ensuring that the teacher target for a real row is produced from a context excluding that row. Unless stated otherwise, all real-data teacher predictions below use this OOF construction. The resulting real-only OOF mixed baseline, which uses hard labels and OOF teacher predictions on real rows only, is

$$\widehat{\theta}_R \in \arg\min_{\theta\in\Theta} \widehat{L}_{\lambda,n}(\theta). \tag{3}$$

2.3 Synthetic Covariate Augmentation

Synthetic data offer a natural means of alleviating data scarcity and can improve learning through pretraining and fine-tuning (Bühler et al., 2026). GEAR uses generated rows more conservatively: they provide additional covariate locations for querying the frozen teacher. To avoid bias from generator-provided labels, Stage 1 discards generated labels and uses only generated covariates with teacher soft targets. Concretely, we draw covariates from a tabular generator, such as a GAN or diffusion model (Xu et al., 2019; Shi et al., 2025a), and never treat them as labeled target examples.

Let $\widetilde{D}_M = \{\widetilde{X}_j\}_{j=1}^M$, where $\widetilde{X}_j \overset{\text{i.i.d.}}{\sim} Q_X$ and Q_X denotes the generated covariate law. For generated queries, we write $T_s(x) = f_T(x \mid D_n)$ for the teacher response. The corresponding population and empirical teacher-imitation objectives are

$$L_s^{Q_X}(\theta) = \mathrm{E}_{\widetilde{X}\sim Q_X}\, \ell_T\{p_\theta(\widetilde{X}), T_s(\widetilde{X})\},$$
$$\widehat{L}_{s,M}^{Q_X}(\theta) = \frac{1}{M}\sum_{j=1}^{M} \ell_T\{p_\theta(\widetilde{X}_j), T_s(\widetilde{X}_j)\}. \tag{4}$$

This stage is most useful when generated volume and fidelity are balanced. A larger M provides more teacher-query locations, whereas a low-fidelity Q_X can shift imitation away from the real covariate law P_X. Section 3 formalizes this trade-off as a finite-query term plus a generator-fidelity floor.

2.4 Two-Stage Distillation

To exploit broad synthetic coverage while preserving real-data anchoring, GEAR adopts a two-stage protocol. Stage 1 pretrains the student using generated covariates and teacher soft targets. Stage 2 then returns to the real table and optimizes an OOF mixed objective that combines hard labels with OOF teacher predictions, correcting the Stage 1 student toward the target joint distribution P. From a transfer-learning perspective, Stage 1 acquires knowledge from auxiliary generated data, whereas Stage 2 performs real-data anchoring in the target domain. Abundant synthetic queries reduce the estimation burden, whereas real samples mitigate distribution shift and align the student with the target distribution (Zhuang et al., 2020; Li et al., 2022).

Given a center θ_0, let $\Theta_2(\theta_0) \subseteq \Theta$ denote the implementation's Stage 2 search region around θ_0. This region may be induced by initialization, early stopping, or regularization. Algorithm 1 in Appendix A summarizes the complete pipeline from generated teacher queries to OOF real-data anchoring.

Stage 1 performs pure teacher imitation on generated covariates:

$$\widehat{\theta}_1 \in \arg\min_{\theta\in\Theta} \widehat{L}_{s,M}^{Q_X}(\theta). \tag{5}$$

Stage 2 starts from $\widehat{\theta}_1$ and returns to the real labeled sample. It optimizes the OOF mixed objective in (2), anchoring the Stage 1 student to real labels and OOF teacher predictions within the implementation's Stage 2 search region:

$$\widehat{\theta}_2 \in \arg\min_{\theta\in\Theta_2(\widehat{\theta}_1)} \widehat{L}_{\lambda,n}(\theta). \tag{6}$$

The restricted Stage 2 search region does not guarantee a lower empirical OOF loss than an unrestricted real-only fit. Instead, initializing at $\widehat{\theta}_1$ provides a teacher-informed starting point that can facilitate optimization toward a favorable target-domain solution relative to random initialization.

3 THEORETICAL GUARANTEE

This section presents a compact risk certificate for the generated-imitation part of GEAR. The baseline is the real-only OOF mixed estimator $\widehat{\theta}_R$ in (3), and the target criterion is the population risk $R(\theta) = \mathrm{E}_P\{\ell(p_\theta(X), Y)\}$. The certificate characterizes when generated-only teacher imitation can outperform the real-only baseline.

We use the empirical objectives $\widehat{L}_{s,M}^{Q_X}$ and $\widehat{L}_{\lambda,n}$ for synthetic teacher imitation and real-data OOF distillation, respectively. The conversion gaps quantify surrogate-to-target bias. The terms ϵ_s and $\epsilon_\lambda^{\mathrm{oof}}$ are finite-query and cross-fitting radii. Here, TV denotes total variation, $B\,\mathrm{TV}(P_X, Q_X)$ is the generator-fidelity floor, and $\underline{R}_R$ is a conservative lower certificate for the real-only baseline. Formal definitions appear in Appendix B.

Theorem 1 in Appendix B.1 gives a sufficient condition for Stage 1 to outperform real-only OOF distillation. The following corollary isolates the query-volume and generator-fidelity terms.

Corollary 1 (Fidelity–volume specialization)**.** *Suppose the generated-query radius satisfies*

$$\epsilon_s(\Theta, M, \delta) \le \mathfrak{C}_s(\Theta, \delta) M^{-1/2} \tag{7}$$

for a class-dependent constant $\mathfrak{C}_s(\Theta, \delta)$. Using the TV control in the supplementary material, a sufficient condition for $R(\widehat{\theta}_1) < R(\widehat{\theta}_R)$ is

$$B\,\mathrm{TV}(P_X, Q_X) + 2\mathfrak{C}_s(\Theta, \delta/2) M^{-1/2} < \underline{R}_R(\delta/2) - \inf_{\theta\in\Theta} L_s^{Q_X}(\theta) - \beta_s(\Theta). \tag{8}$$

Corollary 1 isolates the fidelity–volume price

$$H_1(M) = B\,\mathrm{TV}(P_X, Q_X) + 2\mathfrak{C}_s(\Theta, \delta/2) M^{-1/2}. \tag{9}$$

The $M^{-1/2}$ term decays with generated volume, whereas $B\,\mathrm{TV}(P_X, Q_X)$ is an irreducible fidelity floor intrinsic to the generator. When Q_X closely matches P_X so that $\mathrm{TV}(P_X, Q_X) \approx 0$, increasing M progressively tightens the certificate until it reaches this fidelity floor. This yields a scaling law for synthetic expansion: the certifiable finite-query error decays as $M^{-1/2}$ only while sampling error is the primary bottleneck. When Q_X is poorly matched to P_X, additional queries cannot mitigate the resulting distribution shift.

Stage 2 addresses the source-induced bias inherited from generated-query imitation. In transfer-learning terms, Stage 1 uses source-like synthetic coverage to reduce the estimation burden and provide an informed initialization, whereas Stage 2 uses real labels and OOF teacher predictions to debias the student under the target distribution. This correction is essential for controlling target risk when $Q_X \neq P_X$ (Cortes & Mohri, 2014; Li et al., 2022).

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate on the TALENT (Liu et al., 2025) and TabArena (Erickson et al., 2026) benchmarks. For all settings, we use ROC-AUC for binary classification and macro one-vs-rest AUC for multiclass classification; reported values are averaged over five random runs. The generated-to-real sample ratio for Stage 1 is swept over $K = M/n \in \{1, 5, 10, 20, 30, 40\}$, where M is the number of generated covariates and n is the real training size.

We use TabICL v2 (Qu et al., 2026), TabPFN v3 (Grinsztajn et al., 2026)[2], and TabDPT v1.2 (Ma et al., 2024b) as teacher models. The default student is an MLP, and the student-family ablation evaluates LightGBM and XGBoost instead. We additionally benchmark against three strong supervised tabular baselines: CatBoost, RealMLP (Holzmüller et al., 2024), and TabM (Gorishniy et al., 2025). The primary Stage 1 generator is TabPFGen (Ma et al., 2024a), an energy-based in-context tabular generator that conditions on the observed table and produces synthetic covariates without fitting a separate task-specific generator. We also evaluate Copula, CTGAN (Xu et al., 2019), and

[2]TabPFN v3 is used only for scientific research.

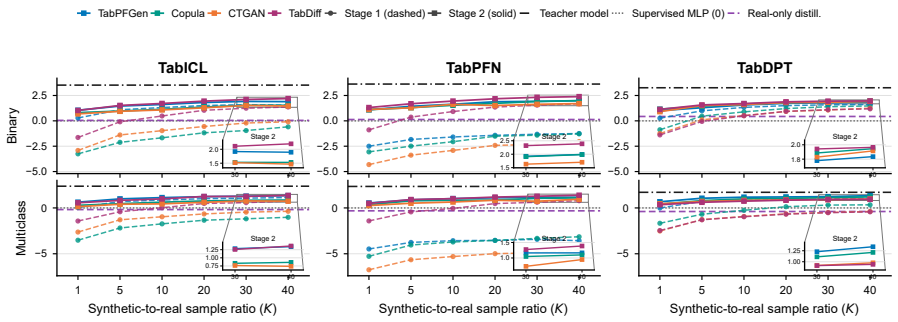


Figure 3: Generator trajectories on TALENT. Columns correspond to TabICL, TabPFN, and TabDPT, and rows to binary and multiclass tasks. Curves show mean gains in AUC points over the supervised MLP across generated-to-real ratios K. Colors denote generators; dashed circles and solid squares indicate Stage 1 and Stage 2, respectively. Horizontal lines mark the teacher, supervised-MLP, and real-only-distillation references.

TabDiff (Shi et al., 2025a), covering a dependence-preserving statistical generator, a GAN-based generator, and a diffusion-based generator.

We compare the following procedures. The *teacher model* is the frozen TFM evaluated directly with the real training table as context. The *supervised student* is trained only on real data with hard labels. The *synthetic-pretraining student* is trained on generated covariates with soft teacher targets in Stage 1. The *real-data distillation baseline* uses only real rows with out-of-fold teacher predictions and hard labels (Tanna et al., 2026b). The *two-stage student (GEAR)* in Section 2.4 starts from synthetic pretraining and then applies the real-data anchoring step.

Implementation and efficiency. The default student is a residual MLP with LayerNorm and SiLU activations, hidden width 512, four blocks, and dropout 0.2; Stage 2 uses a soft-label weight of $\lambda = 0.7$. Timing experiments use a server with a single NVIDIA A800-SXM GPU and two Intel Xeon Platinum 8369B CPUs. We time direct TFM inference on the GPU and student inference on the CPU and, where supported, on the same GPU. Detailed optimization settings and implementation are provided in Appendix E.3.

4.2 Main Synthetic-Volume Study

Q1: Does distillation improve a deployable MLP? Figure 3 compares four generators on matched datasets and reports AUC-point gains relative to the supervised MLP; thus, zero denotes the supervised baseline, whereas the real-data distillation line represents the performance attainable without generated covariates. Across teacher–task panels, the trajectories exhibit a consistent pattern: increasing K improves Stage 1 until the gains plateau at a generator-dependent level, while Stage 2 generally improves the corresponding Stage 1 solution through real-data anchoring.

For TabPFGen at $K = 40$, the two-stage MLP increases mean AUC by 1.91 points on binary tasks and 1.27 points on multiclass tasks relative to the supervised MLP. As K increases, the marginal benefit of additional generated rows diminishes. From $K = 1$ to $K = 10$, mean Stage 1 AUC rises by 0.98 points on binary tasks and 0.86 points on multiclass tasks, whereas the increase from $K = 10$ to $K = 40$ is only 0.29 and 0.17 points, respectively. This saturation is consistent with the volume–fidelity trade-off: additional teacher queries reduce finite-sample error, whereas real-data anchoring is required to mitigate the generated–real distribution mismatch.

Q2 (Stage 1 Analysis): How much does the generator matter? Generator choice materially affects both the Stage 1 plateau and the benefit available from Stage 2.

TabPFGen and TabDiff produce richer energy-based and diffusion-based query distributions. Their stronger Stage 1 performance is often accompanied by larger two-stage gains. Copula is a useful low-capacity stress test because it preserves a simple dependence structure, but it is not designed to

Table 1: GEAR MLP versus CatBoost at $K = 40$. Cells report mean $\Delta_{\mathrm{GEAR-CB}}$ / two-sided Wilcoxon p-value.

Task	TabPFGen	Copula	CTGAN	TabDiff
Bin.	$+0.87 / 0.04$	$+0.81 / 0.19$	$+0.71 / 0.10$	$+1.18 / < 0.01$
Multi.	$+0.26 / 0.05$	$-0.05 / 0.08$	$-0.17 / 0.28$	$+0.18 / 0.02$

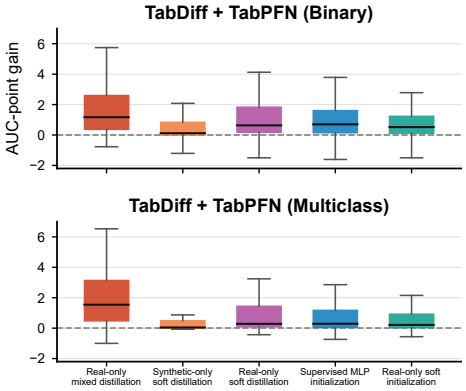


Figure 4: Stage 1 replacement ablations using TabPFN as the teacher. Boxplots report paired gains in AUC points of GEAR over five alternatives, ordered from real-only mixed distillation to real-only soft initialization.

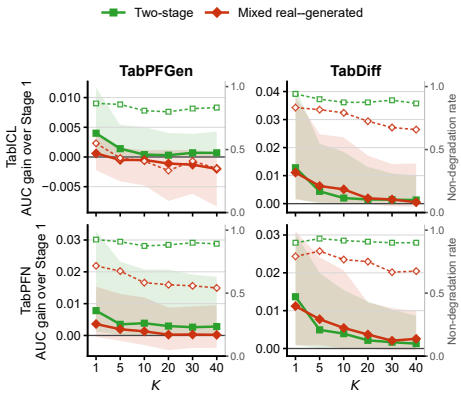


Figure 5: Two-stage versus mixed real–generated distillation on TALENT binary tasks. Rows denote TFM teachers, and columns denote TabPFGen and TabDiff. Solid curves show paired median gains in AUC points over Stage 1 with interquartile bands; dashed hollow-marker curves show non-degradation rates relative to Stage 1 (right axis).

learn complex structural dependencies or causal relationships. CTGAN typically lies between these extremes. As K grows, Stage 1 with higher-fidelity generators can surpass the real-only reference, although weaker generators need more generated rows to reach the same range. By contrast, in the TabICL and TabPFN panels, Copula often remains below real-only distillation even with larger generated pools. This pattern matches the fidelity floor in Corollary 1.

Stage 2 usually improves its corresponding Stage 1 curve because it re-optimizes on real rows, but it is not an automatic repair step. If synthetic teacher imitation pushes the student toward a low-performing region, OOF correction on real rows has limited room to recover. Generator quality therefore affects both Stage 1 accuracy and the usefulness of subsequent real-data anchoring. At $K = 40$, Table 1 reports mean paired AUC-point differences and two-sided Wilcoxon p-values after averaging available teacher results per task. The fractions of positive paired differences are 57.7–69.8% (binary) and 60.9–64.4% (multiclass); TabDiff shows the broadest positive shift, whereas the Copula and CTGAN multiclass means are negative despite positive medians.

Q3 (Stage 1 Analysis): Do generated teacher queries really help? One possibility is that Stage 2 benefits only from the initialization supplied by pretraining or from a longer training schedule. We therefore retain the same two-stage protocol and compare GEAR with five alternatives: *real-only mixed distillation*, *synthetic-only soft distillation*, *real-only soft distillation*, *supervised MLP initialization*, and *real-only soft initialization*. The first three use, respectively, real rows with mixed soft and hard targets, generated rows with soft teacher targets only, and real rows with soft teacher targets only. The final two replace synthetic pretraining with a supervised MLP warm start or a real-only soft-distillation warm start before the same real-data second stage. If synthetic pretraining merely provided an arbitrary initialization or if the gain came solely from a longer schedule, these alternatives should perform comparably. Figure 4 shows a representative comparison with TabPFN as the teacher. The full results appear in Appendix E.5.

Table 2: Tree-student distillation with TabPFGen. Values are paired gains in AUC points over supervised training; subscripts denote standard errors.

Teacher	Task	Student	S1 Δ	Real Δ	S2 Δ
TabICL	Bin.	LGBM	$+0.79_{(0.25)}$	$+0.92_{(0.24)}$	$\mathbf{+1.53}_{(0.20)}$
		XGB	$+0.96_{(0.21)}$	$+1.11_{(0.15)}$	$\mathbf{+1.37}_{(0.18)}$
	Multi.	LGBM	$-6.34_{(1.14)}$	$-8.69_{(1.04)}$	$\mathbf{-5.93}_{(1.14)}$
		XGB	$+0.22_{(0.22)}$	$+0.40_{(0.13)}$	$\mathbf{+0.51}_{(0.12)}$
TabPFN	Bin.	LGBM	$-2.57_{(0.76)}$	$+0.91_{(0.24)}$	$\mathbf{+1.32}_{(0.20)}$
		XGB	$-2.71_{(0.90)}$	$+1.14_{(0.16)}$	$\mathbf{+1.38}_{(0.18)}$
	Multi.	LGBM	$-10.39_{(1.62)}$	$-8.96_{(1.12)}$	$\mathbf{-6.02}_{(1.22)}$
		XGB	$-5.28_{(1.45)}$	$+0.15_{(0.25)}$	$\mathbf{+0.44}_{(0.12)}$
TabDPT	Bin.	LGBM	$+0.90_{(0.24)}$	$+0.65_{(0.27)}$	$\mathbf{+1.11}_{(0.23)}$
		XGB	$+0.69_{(0.20)}$	$+0.82_{(0.20)}$	$\mathbf{+0.98}_{(0.18)}$
	Multi.	LGBM	$-2.50_{(0.55)}$	$\mathbf{-1.77}_{(0.42)}$	$-2.17_{(0.48)}$
		XGB	$-0.36_{(0.29)}$	$-0.20_{(0.26)}$	$\mathbf{-0.11}_{(0.22)}$

The results answer Q3 affirmatively. For TabDiff with a TabPFN teacher on binary tasks, GEAR improves over real-only mixed distillation by 2.61 AUC points and wins on 92.6% of the paired datasets. It also exceeds supervised MLP initialization by 1.46 points. On multiclass tasks, GEAR gains 2.38 points over real-only mixed distillation and wins on 94.2% of the paired datasets. Comparisons with supervised MLP initialization and real-only soft initialization show that neither a generic warm start nor additional training explains the gains. Synthetic pretraining yields stronger Stage 2 performance under comparable budgets. Moreover, real-only soft distillation underperforms synthetic-only soft distillation, indicating that generated covariates expand teacher-query coverage rather than merely repeating distillation on observed rows. Thus, the generated stage supplies informative query locations, while the real stage restores target-domain alignment.

Q4: Does GEAR generalize across student families? This study replaces the MLP student with LightGBM and XGBoost while retaining the same two-stage logic. The tree-based student study uses TabPFGen as its generator: Stage 1 first trains the tree student on generated teacher targets, and Stage 2 continues training from that Stage 1 student using real-data anchoring. Table 2 separates three paths: Stage 1-only synthetic teacher imitation, real-data distillation from scratch, and the full two-stage route.

Because tree models are non-differentiable and poorly suited to fine-tuning, distillation is challenging; nevertheless, two-stage training improves LightGBM by 1.11–1.53 AUC points and XGBoost by 0.98–1.38 points over supervised training across all three teachers. Aggregated across datasets, these students also surpass CatBoost, the strongest non-TFM baseline, by 0.74 and 0.84 AUC points, respectively, showing that the gains extend beyond matched supervised comparisons. XGBoost is especially stable on multiclass tasks, with positive gains for TabICL and TabPFN and near parity for TabDPT. LightGBM is less stable on multiclass tasks, where all three distillation paths remain below supervised training, consistent with Tanna et al. (2026a). Nevertheless, for TabICL and TabPFN, two-stage training is less harmful than real-only distillation. Overall, the benefit is student-dependent. Generated teacher queries can guide a useful learning trajectory, whereas the benefit of real-data anchoring depends strongly on the student family and task type. We also evaluate TabM and MLPs with one to three blocks; shallower MLPs show a larger two-stage advantage. Complete results are provided in Appendices E.6 and E.7.

4.3 Ablation Studies

Staged optimization versus mixed training. This ablation tests whether the generated and real objectives can be optimized jointly. The mixed real–generated baseline combines generated teacher targets with OOF real-data anchoring but does not separate query expansion from real-data anchoring. Figure 5 reports paired results across K on TALENT binary tasks, including median AUC

gains and non-degradation rates relative to Stage 1; each rate is the fraction of paired tasks where the corresponding sequential or mixed method is at least as accurate as Stage 1.

Figure 5 shows representative TabPFGen and TabDiff comparisons; the full sweep appears in Appendix E.1. With TabPFGen, two-stage training has higher median gains in AUC points than mixed training for both teachers at every K, while the mixed approach approaches or falls below Stage 1 as K grows. Mixed training can be competitive for other generators, but two-stage training achieves non-degradation rates that are 4.2–48.7 percentage points higher in every comparison. Separating query expansion from real-data anchoring therefore improves stability.

Soft-label mixing. We examine GEAR's sensitivity to the weighting parameter λ, with full results reported in Appendix E.2. In Stage 1, pure or near-pure teacher supervision outperforms training with generated hard labels, confirming that generated rows are more useful as teacher-query locations than as labeled examples. In Stage 2, sensitivity to λ is weaker but non-negligible: teacher soft targets remain beneficial when balanced with real labels, thereby anchoring the objective to the target distribution.

5 Related Work

Tabular foundation models. Gradient-boosted trees and tuned neural architectures remain strong tabular baselines (Chen & Guestrin, 2016; Ke et al., 2017; Gorishniy et al., 2021; Grinsztajn et al., 2022). TabPFN introduced context-based prediction from a labeled table (Hollmann et al., 2022); subsequent TFMs improve pretraining strategies, extend context length, or incorporate more structured-data modeling (Ma et al., 2024b; Qu et al., 2026; Grinsztajn et al., 2026; Wang et al., 2026). Scalability methods retrieve, adapt, chunk, or optimize the context supplied to a TFM (Thomas et al., 2024; Liu & Ye, 2025; Sergazinov & Yin, 2025; Chen et al., 2026).

Distilling tabular foundation models. Knowledge distillation compresses powerful predictors into deployable models (Buciluǎ et al., 2006; Hinton et al., 2015). TFM-specific approaches include in-context data distillation, MotherNet, TabDistill, end-to-end compression, real-data-only distillation, and interaction distillation (Ma et al., 2024c; Müller et al., 2025; Dissanayake & Dutta, 2025; Zabërgja et al., 2026; Tanna et al., 2026b; Jia et al., 2026). GEAR's distinguishing contribution is the separation of generated teacher-query expansion from real-data anchoring, which connects it to sample-bias correction under distribution shift (Cortes & Mohri, 2014).

Synthetic tabular data generation. Synthetic tabular generators include CTGAN, diffusion models, and TFM-based generators (Xu et al., 2019; Kotelnikov et al., 2023; Zhang et al., 2024; Shi et al., 2025a; Ma et al., 2024a; Jiang et al., 2026b). Their utility depends on generator quality, task type, and evaluation protocol (Shi et al., 2025b; Challagundla et al., 2025); accordingly, GEAR uses synthetic rows as teacher-query locations rather than as labeled target examples.

6 Conclusion

We present GEAR, a modular two-stage framework that distills in-context tabular foundation models into efficient, context-free predictors through generative query expansion and real-data anchoring. In the main MLP experiments, GEAR improves over supervised learning and real-data-only distillation while substantially reducing TFM inference costs. Our risk certificate further explains the trade-off between query volume and generator fidelity.

This work focuses on classification and single-teacher distillation; GEAR could be extended to regression and multi-teacher distillation. Promising directions include fidelity-aware query generation and combining heterogeneous generators. Another direction is to introduce adversarial distribution alignment during distillation, encouraging generated queries to expand teacher coverage while remaining close to the real-data distribution.

AI USE STATEMENT

We used ChatGPT 5.6 to assist with literature searches, language polishing, grammar checking, software implementation, and code drafting. All AI-assisted code was reviewed, tested, and verified. All experiments and scientific analyses were conducted by the authors, and generative AI was not used to generate experimental results. We reviewed all AI-assisted content and take responsibility for the final manuscript.

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.

Magnus Bühler, Lennart Purucker, and Frank Hutter. Causal data augmentation for robust fine-tuning of tabular foundation models. *arXiv preprint arXiv:2601.04110*, 2026.

Raju Challagundla, Mohsen Dorodchi, Pu Wang, and Minwoo Lee. Synthetic tabular data generation: A comparative survey for modern techniques. *arXiv preprint arXiv:2507.11590*, 2025.

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.

Yilong Chen, Xueying Ding, and Leman Akoglu. Vip-cop: Context optimization for tabular foundation models. *arXiv preprint arXiv:2605.12904*, 2026.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

Pasan Dissanayake and Sanghamitra Dutta. Tabdistill: Distilling transformers into neural nets for few-shot tabular classification. *arXiv preprint arXiv:2511.05704*, 2025.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2026.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

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.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35: 507–520, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin, 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. *arXiv preprint arXiv:2207.01848*, 2022.

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

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned mlps and boosted trees on tabular data. *Advances in Neural Information Processing Systems*, 37:26577–26658, 2024.

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.

Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and Han-Jia Ye. Representation learning for tabular data: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026a.

Xiangjian Jiang, Mingxuan Liu, Nikola Simidjievski, Tassilo Klein, and Mateja Jamnik. Tabular foundation model for generative modelling. *arXiv preprint arXiv:2605.09424*, 2026b.

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.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pp. 17564–17579. PMLR, 2023.

Sai Li, T Tony Cai, and Hongzhe Li. Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):149–173, 2022.

Yijun Lin and Sai Li. Context-constrained transfer learning for tabular foundation models via data distillation. *arXiv preprint arXiv:2607.04809*, 2026.

Si-Yang Liu and Han-Jia Ye. Tabpfn unleashed: A scalable and effective solution to tabular classification problems. *arXiv preprint arXiv:2502.02527*, 2025.

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.

Junwei Ma, Apoorv Dankar, George Stein, Guangwei Yu, and Anthony Caterini. Tabpfgen–tabular data generation with tabpfn. *arXiv preprint arXiv:2406.05216*, 2024a.

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. *arXiv preprint arXiv:2410.18164*, 2024b.

Junwei Ma, Valentin Thomas, Guangwei Yu, and Anthony Caterini. In-context data distillation with tabpfn. *arXiv preprint arXiv:2402.06971*, 2024c.

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.

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.

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*, 2025.

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.

Renat Sergazinov and Shao-An Yin. Chunked tabpfn: Exact training-free in-context learning for long-context tabular data. *arXiv preprint arXiv:2509.00326*, 2025.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. Tabdiff: a mixed-type diffusion model for tabular data generation. In *International Conference on Learning Representations*, volume 2025, pp. 37353–37375, 2025a.

Ruxue Shi, Yili Wang, Mengnan Du, Xu Shen, Yi Chang, and Xin Wang. A comprehensive survey of synthetic tabular data generation. *arXiv preprint arXiv:2504.16506*, 2025b.

Aditya Tanna, Nassim Bouarour, Mohamed Bouadi, Vinay Kumar Sankarapu, and Pratinav Seth. Distilling tabular foundation models for structured health data. *arXiv preprint arXiv:2605.18702*, 2026a.

Aditya Tanna, Nassim Bouarour, Mohamed Bouadi, Pratinav Seth, et al. Pocket foundation models: Distilling tfms into cpu-ready gradient-boosted trees. *arXiv preprint arXiv:2605.18654*, 2026b.

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.

Boris Van Breugel and Mihaela Van Der Schaar. Why tabular foundation models should be a research priority. *arXiv preprint arXiv:2405.01147*, 2024.

Yuanrui Wang, Xingxuan Zhang, Han Yu, Mingchao Ming, Gang Ren, Hao Yuan, Li Mao, Yunjia Zhang, Chun Yuan, and Peng Cui. Limix-2m: Mitigating low-rank collapse and attention bottlenecks in tabular foundation models. *arXiv preprint arXiv:2606.04485*, 2026.

Derek Xu, Olcay Cirit, Reza Asadi, Yizhou Sun, and Wei Wang. Mixture of in-context prompters for tabular pfns. In *International Conference on Learning Representations*, volume 2025, pp. 24630–24665, 2025.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Harry Chao. A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. *Advances in Neural Information Processing Systems*, 38: 135605–135637, 2026.

Guri Zabërgja, Rafiq Kamel, Arlind Kadra, Christian MM Frey, and Josif Grabocka. End-to-end compression for tabular foundation models. *arXiv preprint arXiv:2602.05649*, 2026.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *International Conference on Learning Representations*, volume 2024, pp. 52829–52857, 2024.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76, 2020.

Reproducibility Statement

Algorithm 1 specifies the two-stage distillation procedure. Section 4.1 describes the evaluation setup, and Appendix E.3 provides implementation and efficiency details. Appendices B–D give the assumptions, definitions, and proofs supporting Section 3. Additional ablations, complete tree-student results, and the numerical data for Figure 1 are provided in Appendix E.

Appendix Map

Appendix A gives the complete distillation algorithm. Sections B–D provide the formal definitions, bias bounds, and proofs. Section E reports the supplementary ablations, implementation and efficiency details, complete tree-student results, and the numerical values underlying Figure 1.

A Complete Distillation Algorithm

Algorithm 1 Two-stage generative distillation with OOF real-data anchoring.

Input: real table $D_n = \{(X_i, Y_i)\}_{i=1}^n$, frozen teacher f_T, student $f_S(\cdot;\theta)$.
Input: generated pool size M, fold count K_{cf}, losses ℓ, ℓ_T, weight λ, Stage 2 search rule Θ_2.
Output: two-stage student $\widehat{\theta}_2$.
Stage 1: generated teacher imitation.
Draw generated covariates $\widetilde{X}_1, \ldots, \widetilde{X}_M \sim Q_X$.
for $j = 1, \ldots, M$ **do**
    Query $f_T(\widetilde{X}_j \mid D_n)$.
end for
$\widehat{\theta}_1 \leftarrow \arg\min_{\theta\in\Theta} \widehat{L}_{s,M}^{Q_X}(\theta)$.
Stage 2: OOF real-data anchoring.
Partition D_n into folds $I_1, \ldots, I_{K_{\mathrm{cf}}}$.
for $k = 1, \ldots, K_{\mathrm{cf}}$ **do**
    Let $D_n^{(-k)} = D_n \setminus \{(X_i, Y_i) : i \in I_k\}$.
    for each $i \in I_k$ **do**
        Query $f_T(X_i \mid D_n^{(-k)})$.
    end for
end for
Define $\widehat{L}_{\lambda,n}$ by mixing real-label and OOF teacher terms as in (2).
Initialize at $\widehat{\theta}_1$ and optimize the OOF mixed objective within $\Theta_2(\widehat{\theta}_1)$:
    $\widehat{\theta}_2 \leftarrow \arg\min_{\theta\in\Theta_2(\widehat{\theta}_1)} \widehat{L}_{\lambda,n}(\theta)$.
return $\widehat{\theta}_2$.

B Technical Definitions for Section 3

Assumption 1 (Conditional sampling and bounded losses)**.** (i) *Conditional on the fitted generator and the real-data context used by the teacher, the generated covariates are i.i.d. draws from Q_X.* (ii) *The OOF partition is fixed in advance or sampled independently of D_n. For each fold k, T_k is measurable with respect to $D_n^{(-k)}$. Conditional on the realized split and $D_n^{(-k)}$, the held-out observations in I_k are i.i.d. draws from P.* (iii) *The losses ℓ, ℓ_T, and ℓ_λ used in the analysis are bounded in $[0, B]$. For cross-entropy, this can be enforced by clipping $p_{\theta,a}(x) \geq \varepsilon$ for all a, x, θ, which yields $B = \log(1/\varepsilon)$.*

For any candidate class $\Theta' \subseteq \Theta$, define the uniform surrogate-to-target conversion gaps

$$\mathfrak{b}_s(\Theta') = \sup_{\theta\in\Theta'} |R(\theta) - L_s^{Q_X}(\theta)|,$$
$$\mathfrak{b}_\lambda(\Theta') = \sup_{\theta\in\Theta'} |R(\theta) - L_\lambda^{P}(\theta)|. \tag{10}$$

For generated-only Stage 1, the conversion gap decomposes into a teacher-surrogate component and a generated-to-real covariate component:

$$\beta_s(\Theta') = \sup_{\theta\in\Theta'} \left|R(\theta) - \mathrm{E}_{P_X}\, \ell_T\{p_\theta(X), T_s(X)\}\right|,$$
$$\Delta_s(\Theta') = \sup_{\theta\in\Theta'} \left|\mathrm{E}_{P_X}\, \ell_T\{p_\theta(X), T_s(X)\} - L_s^{Q_X}(\theta)\right|. \tag{11}$$

Then $\mathfrak{b}_s(\Theta') \le \beta_s(\Theta') + \Delta_s(\Theta')$ by the triangle inequality.

Define the generated-sample and fold-k OOF mixed-loss classes as

$$\mathcal{F}_s(\Theta') = \{x \mapsto \ell_T\{p_\theta(x), T_s(x)\} : \theta \in \Theta'\},$$
$$\mathcal{F}_{\lambda,k}(\Theta') = \{(x,y) \mapsto \ell_\lambda\{p_\theta(x), y, T_k(x)\} : \theta \in \Theta'\}.$$

We use the standard empirical-process notation

$$\mathfrak{R}_m^D(\mathcal{F}) = \mathrm{E}_{Z_{1:m}\sim D,\,\sigma}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{m}\sum_{i=1}^m \sigma_i f(Z_i)\right|\right],$$

where σ_i are independent Rademacher random variables. When a loss class depends on a fitted generator or teacher map, the corresponding complexity is evaluated conditionally after that map is fixed. For generated samples, define

$$\epsilon_s(\Theta', M, \delta) = 2\mathfrak{R}_M^{Q_X}\{\mathcal{F}_s(\Theta')\} + B\sqrt{\frac{\log(2/\delta)}{2M}}. \tag{12}$$

For a fixed class Θ', let $\epsilon_\lambda^{\mathrm{oof}}(\Theta', n, \delta)$ denote a valid uniform deviation radius for the OOF real observations. Under equal-sized folds with $m = n/K_{\mathrm{cf}}$, one valid choice is

$$\epsilon_\lambda^{\mathrm{oof}}(\Theta', n, \delta) = \frac{1}{K_{\mathrm{cf}}}\sum_{k=1}^{K_{\mathrm{cf}}}\left[2\mathfrak{R}_m^P\{\mathcal{F}_{\lambda,k}(\Theta')\} + B\sqrt{\frac{\log(2K_{\mathrm{cf}}/\delta)}{2m}}\right]. \tag{13}$$

Here $\mathfrak{R}_m^P\{\mathcal{F}_{\lambda,k}(\Theta')\}$ denotes the conditional Rademacher complexity after the fold-k teacher map has been fixed by $D_n^{(-k)}$.

Because $\widehat{\theta}_R$ is data dependent, a high-probability lower certificate for its target risk is useful. A random quantity $\underline{R}_R(\delta)$ is a $(1-\delta)$ lower certificate if

$$\Pr\{\underline{R}_R(\delta) \le R(\widehat{\theta}_R)\} \ge 1 - \delta. \tag{14}$$

One valid theoretical certificate is

$$\underline{R}_R(\delta) = \widehat{L}_{\lambda,n}(\widehat{\theta}_R) - \epsilon_\lambda^{\mathrm{oof}}(\Theta, n, \delta) - \mathfrak{b}_\lambda(\Theta). \tag{15}$$

B.1 Generated Stage 1 Comparison Certificate

Theorem 1 (Generated Stage 1 certificate)**.** *Under Assumption 1, with probability at least* $1-\delta$*,*

$$R(\widehat{\theta}_1) - R(\widehat{\theta}_R) \le \widehat{L}_{s,M}^{Q_X}(\widehat{\theta}_1) - \widehat{L}_{\lambda,n}(\widehat{\theta}_R) + \epsilon_s(\Theta, M, \delta/2) + \epsilon_\lambda^{\mathrm{oof}}(\Theta, n, \delta/2) + \mathfrak{b}_s(\Theta) + \mathfrak{b}_\lambda(\Theta). \tag{16}$$

When the right-hand side of (16) is negative, the certificate guarantees that generated-only Stage 1 outperforms the real-only baseline in target risk. The theorem separates two factors taht determine whether additional generated rows help. More queries reduce the finite-query price, but they do not remove a persistent mismatch between generated and real covariates. The following specialization makes this volume–fidelity trade-off explicit.

C Bias Decomposition and Transfer Bounds

The quantities in (10) are uniform conversion gaps between the target risk $R(\theta)$ and the surrogate objective optimized at each stage. For any $\Theta' \subseteq \Theta$, the Stage 1 gap satisfies

$$\begin{aligned}\mathfrak{b}_s(\Theta') \le \sup_{\theta\in\Theta'} \big| R(\theta) - \mathrm{E}_{P_X}\, \ell_T\{p_\theta(X), T_s(X)\}\big| \\ + \sup_{\theta\in\Theta'} \big| \mathrm{E}_{P_X}\, \ell_T\{p_\theta(X), T_s(X)\} \\ - \mathrm{E}_{Q_X}\, \ell_T\{p_\theta(\widetilde{X}), T_s(\widetilde{X})\}\big|. \end{aligned} \tag{17}$$

The first term quantifies teacher-surrogate bias on real covariates and increases when the teacher's soft labels poorly approximate the true supervised loss. The second term is $\Delta_s(\Theta')$ from (11); it quantifies generated-to-real fidelity bias and can increase because of out-of-support generated observations, mode collapse, subgroup undercoverage, or teacher miscalibration in generated regions.

If $g_\theta(x) = \ell_T\{p_\theta(x), T_s(x)\} \in [0, B]$, then

$$\sup_{\theta\in\Theta'} \Big| \mathrm{E}_{P_X}\, g_\theta(X) - \mathrm{E}_{Q_X}\, g_\theta(\widetilde{X})\Big| \le B\, \mathrm{TV}(P_X, Q_X),$$

where $\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)| = \frac{1}{2}\|P - Q\|_1$ for dominated probability laws.

The OOF mixed conversion gap $\mathfrak{b}_\lambda(\Theta')$ is analogous but is evaluated on real observations and therefore does not include generated-to-real covariate shift. When $\ell_\lambda = (1-\lambda)\ell + \lambda\ell_T$ uses the same hard-label loss ℓ as R,

$$\begin{aligned}\mathfrak{b}_\lambda(\Theta') \le \lambda \sup_{\theta\in\Theta'} \Big| \mathrm{E}_P\, \ell\{p_\theta(X), Y\} \\ - \frac{1}{K_{\mathrm{cf}}} \sum_{k=1}^{K_{\mathrm{cf}}} \mathrm{E}_P\, \ell_T\{p_\theta(X), T_k(X)\}\Big|. \end{aligned}$$

For the standard cross-entropy instantiation, this abstract conversion gap admits a direct bound in terms of teacher-posterior error.

Lemma 1 (Cross-entropy soft-to-hard transfer)**.** *Assume* $\ell(p, y) = -\log p_y$, $\ell_T(p, q) = -\sum_{a=1}^{L} q_a \log p_a$, *and the clipped student probabilities used in the loss satisfy* $p_{\theta,a}(x) \ge \varepsilon$ *for all* a, x, θ. *Let* $\eta(x) = P(Y = \cdot \mid X = x)$. *Then, for every realized collection of teacher maps and every* $\Theta' \subseteq \Theta$,

$$\begin{aligned}\beta_s(\Theta') &\le \log(1/\varepsilon)\, \mathrm{E}_{P_X}\, \|\eta(X) - T_s(X)\|_1, \\ \mathfrak{b}_\lambda(\Theta') &\le \lambda \log(1/\varepsilon) \frac{1}{K_{\mathrm{cf}}} \sum_{k=1}^{K_{\mathrm{cf}}} \mathrm{E}_{P_X}\, \|\eta(X) - T_k(X)\|_1. \end{aligned}$$

Proof. For any fixed probability vector p whose coordinates are at least ε and any $u, v \in \Delta_{L-1}$,

$$\left| -\sum_a u_a \log p_a + \sum_a v_a \log p_a \right| \le \log(1/\varepsilon)\|u - v\|_1.$$

Applying this inequality with $u = \eta(X)$ and $v = T_s(X)$ yields the bound on β_s. Applying it fold by fold with $v = T_k(X)$, averaging over k, and multiplying by the soft-loss weight λ yields the bound on $\mathfrak{b}_\lambda$. □

D Proofs for Section 3

Lemma 2 (Generated-query deviation)**.** *Under Assumption 1, for any fixed class* $\Theta' \subseteq \Theta$, *with probability at least* $1 - \delta$,

$$\begin{aligned}\sup_{\theta\in\Theta'} \Big| \widehat{L}^{Q_X}_{s,M}(\theta) - L^{Q_X}_s(\theta)\Big| \\ \le 2\mathfrak{R}^{Q_X}_M\{\mathcal{F}_s(\Theta')\} + B\sqrt{\frac{\log(2/\delta)}{2M}} = \epsilon_s(\Theta', M, \delta). \end{aligned}$$

Proof. Condition on the fitted generator, the real-data context used by T_s, and the induced distribution Q_X. The teacher map T_s and the loss class $\mathcal{F}_s(\Theta')$ are then fixed, while the generated queries remain i.i.d. from Q_X. Let

$$Z_+ = \sup_{\theta\in\Theta'}\{\widehat{L}_{s,M}^{Q_X}(\theta) - L_s^{Q_X}(\theta)\},$$
$$Z_- = \sup_{\theta\in\Theta'}\{L_s^{Q_X}(\theta) - \widehat{L}_{s,M}^{Q_X}(\theta)\}.$$

Changing one generated observation changes either functional by at most B/M. McDiarmid's inequality and a union bound therefore imply, with conditional probability at least $1-\delta$,

$$Z_+ \le \mathrm{E}\, Z_+ + B\sqrt{\frac{\log(2/\delta)}{2M}},$$
$$Z_- \le \mathrm{E}\, Z_- + B\sqrt{\frac{\log(2/\delta)}{2M}}.$$

Conditional symmetrization yields

$$\mathrm{E}\, Z_+,\ \mathrm{E}\, Z_- \le 2\mathfrak{R}_M^{Q_X}\{\mathcal{F}_s(\Theta')\}.$$

Because the target deviation equals $\max\{Z_+, Z_-\}$, the conditional claim follows. Averaging over the fitted generator and teacher context yields the stated unconditional result. □

Lemma 3 (Fold-wise OOF deviation)**.** *Under Assumption 1, for any fixed class $\Theta' \subseteq \Theta$, with probability at least $1-\delta$,*

$$\sup_{\theta\in\Theta'}\left|\widehat{L}_{\lambda,n}(\theta) - L_\lambda^P(\theta)\right| \le \epsilon_\lambda^{\mathrm{oof}}(\Theta', n, \delta).$$

The explicit radius in (13) *is valid under the equal-fold convention $|I_k| = m = n/K_{\mathrm{cf}}$.*

Proof. For fold k, condition on the realized split and on $D_n^{(-k)}$. This conditioning fixes the teacher map $T_k(\cdot)$ and the class $\mathcal{F}_{\lambda,k}(\Theta')$, while the held-out observations in I_k remain i.i.d. draws from P. Although $T_k(X_i)$ depends on the query X_i, the map itself is fixed. Define

$$\widehat{L}_{\lambda,k}(\theta) = \frac{1}{m}\sum_{i\in I_k}\ell_\lambda\{p_\theta(X_i), Y_i, T_k(X_i)\},$$
$$L_{\lambda,k}^P(\theta) = \mathrm{E}_{(X,Y)\sim P}\,\ell_\lambda\{p_\theta(X), Y, T_k(X)\},$$

where the population expectation is over a fresh draw from P. Under equal folds,

$$\widehat{L}_{\lambda,n}(\theta) = \frac{1}{K_{\mathrm{cf}}}\sum_{k=1}^{K_{\mathrm{cf}}}\widehat{L}_{\lambda,k}(\theta),$$
$$L_\lambda^P(\theta) = \frac{1}{K_{\mathrm{cf}}}\sum_{k=1}^{K_{\mathrm{cf}}}L_{\lambda,k}^P(\theta).$$

Applying the same conditional symmetrization and bounded-differences argument as in Lemma 2, with confidence level $1-\delta/K_{\mathrm{cf}}$, gives

$$\sup_{\theta\in\Theta'}\left|\widehat{L}_{\lambda,k}(\theta) - L_{\lambda,k}^P(\theta)\right| \le 2\mathfrak{R}_m^P\{\mathcal{F}_{\lambda,k}(\Theta')\} + B\sqrt{\frac{\log(2K_{\mathrm{cf}}/\delta)}{2m}}.$$

A union bound over folds makes all fold-wise bounds hold simultaneously with probability at least $1-\delta$. On this event,

$$\sup_{\theta\in\Theta'}\left|\widehat{L}_{\lambda,n}(\theta) - L_\lambda^P(\theta)\right| \le \frac{1}{K_{\mathrm{cf}}}\sum_{k=1}^{K_{\mathrm{cf}}}\sup_{\theta\in\Theta'}\left|\widehat{L}_{\lambda,k}(\theta) - L_{\lambda,k}^P(\theta)\right|,$$

which is exactly (13). □

Lemma 4 (Baseline lower certificate)**.** *Let* $\underline{R}_R(\delta)$ *be defined by* (15)*. Then*

$$\Pr\{\underline{R}_R(\delta) \le R(\widehat{\theta}_R)\} \ge 1 - \delta.$$

Proof. Apply Lemma 3 with $\Theta' = \Theta$. Because the event is uniform over the fixed class Θ, it may be evaluated at the data-dependent output $\widehat{\theta}_R$; no independence between $\widehat{\theta}_R$ and the OOF sample is required. Thus, with probability at least $1 - \delta$,

$$L_\lambda^P(\widehat{\theta}_R) \ge \widehat{L}_{\lambda,n}(\widehat{\theta}_R) - \epsilon_\lambda^{\mathrm{oof}}(\Theta, n, \delta).$$

By the definition of $\mathfrak{b}_\lambda(\Theta)$,

$$R(\widehat{\theta}_R) \ge L_\lambda^P(\widehat{\theta}_R) - \mathfrak{b}_\lambda(\Theta).$$

Combining the two inequalities establishes (15) as a valid lower certificate. □

Proof of Theorem 1. Let

$$\mathcal{E}_s = \left\{ \sup_{\theta \in \Theta} |\widehat{L}_{s,M}^{Q_X}(\theta) - L_s^{Q_X}(\theta)| \le \epsilon_s(\Theta, M, \delta/2) \right\},$$

and let $\mathcal{E}_R = \{\underline{R}_R(\delta/2) \le R(\widehat{\theta}_R)\}$. By Lemmas 2 and 4, $\Pr(\mathcal{E}_s \cap \mathcal{E}_R) \ge 1 - \delta$.

On $\mathcal{E}_s$,

$$\begin{aligned} R(\widehat{\theta}_1) &\le L_s^{Q_X}(\widehat{\theta}_1) + \mathfrak{b}_s(\Theta) \\ &\le \widehat{L}_{s,M}^{Q_X}(\widehat{\theta}_1) + \epsilon_s(\Theta, M, \delta/2) + \mathfrak{b}_s(\Theta). \end{aligned}$$

Using the explicit lower certificate, on $\mathcal{E}_R$,

$$R(\widehat{\theta}_R) \ge \widehat{L}_{\lambda,n}(\widehat{\theta}_R) - \epsilon_\lambda^{\mathrm{oof}}(\Theta, n, \delta/2) - \mathfrak{b}_\lambda(\Theta).$$

Subtracting the two bounds yields the empirical certificate in (16).

For the oracle form, work on the same event $\mathcal{E}_s \cap \mathcal{E}_R$. For any $\alpha > 0$, choose $\theta_\alpha \in \Theta$ such that

$$L_s^{Q_X}(\theta_\alpha) \le \inf_{\theta \in \Theta} L_s^{Q_X}(\theta) + \alpha.$$

Since $\widehat{\theta}_1$ minimizes $\widehat{L}_{s,M}^{Q_X}$ over Θ,

$$\begin{aligned} L_s^{Q_X}(\widehat{\theta}_1) &\le \widehat{L}_{s,M}^{Q_X}(\widehat{\theta}_1) + \epsilon_s(\Theta, M, \delta/2) \\ &\le \widehat{L}_{s,M}^{Q_X}(\theta_\alpha) + \epsilon_s(\Theta, M, \delta/2) \\ &\le L_s^{Q_X}(\theta_\alpha) + 2\epsilon_s(\Theta, M, \delta/2) \\ &\le \inf_{\theta \in \Theta} L_s^{Q_X}(\theta) + \alpha + 2\epsilon_s(\Theta, M, \delta/2). \end{aligned}$$

Therefore

$$R(\widehat{\theta}_1) \le \inf_{\theta \in \Theta} L_s^{Q_X}(\theta) + \alpha + 2\epsilon_s(\Theta, M, \delta/2) + \mathfrak{b}_s(\Theta).$$

Letting $\alpha \downarrow 0$ and subtracting the lower certificate for $R(\widehat{\theta}_R)$ gives the explicit oracle bound

$$\begin{aligned} R(\widehat{\theta}_1) - R(\widehat{\theta}_R) \le \inf_{\theta \in \Theta} L_s^{Q_X}(\theta) + 2\epsilon_s(\Theta, M, \delta/2) \\ + \mathfrak{b}_s(\Theta) - \underline{R}_R(\delta/2). \end{aligned} \tag{18}$$

If the right-hand side is negative, the target-risk improvement follows on the same high-probability event. No independence between $\mathcal{E}_s$ and $\mathcal{E}_R$ is used; the probability statement follows from a union bound. □

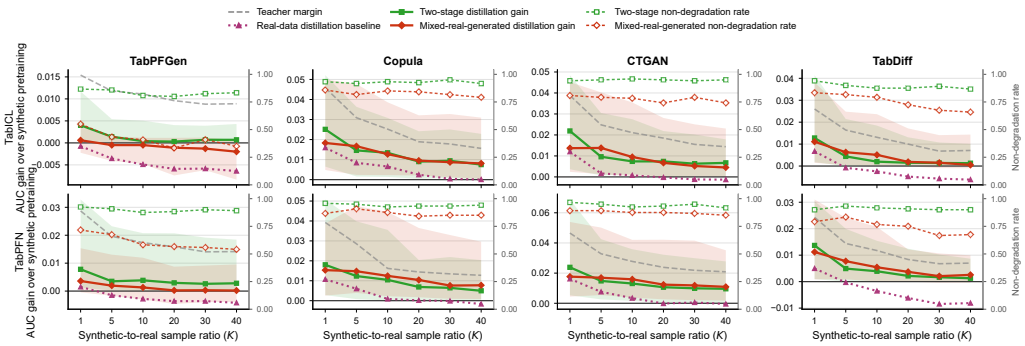


Figure 6: Full comparison of two-stage and mixed real–generated distillation across K on TALENT binary tasks. The panels report paired median AUC gains over synthetic pretraining; shaded bands denote interquartile ranges, and dashed open-marker curves report, for each method, non-degradation rates relative to Stage 1 on the right axis.

Table 3: Soft-label mixing with TabICL on TALENT binary tasks. Stage 1 and Stage 2 entries report mean AUC at the corresponding soft-label weight.

Stage	Metric	0	0.2	0.3	0.5	0.7	0.8	1.0
Stage 1	$K = 10$ mean AUC	0.853	0.856	0.857	0.859	0.859	0.860	**0.860**
	$K = 20$ mean AUC	0.854	0.858	0.859	0.860	0.861	0.862	**0.863**
	$K = 40$ mean AUC	0.854	0.859	0.860	0.861	0.862	0.863	**0.864**
Stage 2	Mean AUC	0.864	0.866	0.866	0.867	0.867	**0.867**	0.867

Proof of Corollary 1. By (11) and the triangle inequality, $\mathfrak{b}_s(\Theta) \le \beta_s(\Theta) + \Delta_s(\Theta)$. The total-variation bound in Appendix C gives $\Delta_s(\Theta) \le B\,\mathrm{TV}(P_X, Q_X)$. Substituting these inequalities and (7) into (18) yields

$$\begin{aligned} R(\widehat{\theta}_1) - R(\widehat{\theta}_R) \le \inf_{\theta\in\Theta} L_s^{Q_X}(\theta) \\ + 2\mathfrak{C}_s(\Theta, \delta/2) M^{-1/2} \\ + \beta_s(\Theta) + B\,\mathrm{TV}(P_X, Q_X) \\ - \underline{R}_R(\delta/2). \end{aligned}$$

Rearranging the condition that the right-hand side be negative gives (8). The stated monotonicity of $H_1(M)$ follows because, for fixed Q_X, its only M-dependent term is proportional to $M^{-1/2}$. The term $B\,\mathrm{TV}(P_X, Q_X)$ is a nonvanishing term in this sufficient certificate; it is an upper bound on the generated-to-real discrepancy, not a lower bound on the actual bias. □

E Additional Experimental Results

E.1 Two-Stage versus Mixed Distillation

E.2 Soft-Label Mixing

E.3 Implementation and Efficiency Details

Because of compliance requirements, we cannot release the experiment code. All experiment-specific hyperparameters, including random seeds, are reported in the supplementary material.

Data preprocessing and teacher construction. We use the fixed training, validation, and test partitions provided by each benchmark. Feature preprocessing is fitted on the real training partition and then applied consistently to validation, test, and generated covariates. All teacher predictions

Table 4: Soft-label mixing with TabPFN on TALENT binary tasks. Stage 1 and Stage 2 entries report mean AUC at the corresponding soft-label weight.

Stage	Metric	0	0.2	0.3	0.5	0.7	0.8	1.0
Stage 1	$K = 10$ mean AUC	0.848	0.849	0.851	**0.856**	0.856	0.855	0.833
	$K = 20$ mean AUC	0.849	0.851	0.852	**0.858**	0.857	0.857	0.835
	$K = 40$ mean AUC	0.849	0.853	0.853	**0.858**	0.858	0.858	0.838
Stage 2	Mean AUC	0.862	0.865	0.865	0.866	0.866	**0.867**	0.865

and student fits use this common feature space. Categorical variables are integer-encoded and missing values are median-imputed before standardization. Standardization uses only the real training partition; columns with standard deviation below 10^{-12} use a unit scale. The teachers are fitted on the normalized real training data with seed 42. TabICL and TabPFN use eight estimator predictions, whereas TabDPT uses four. To avoid self-labeling, Stage 2 uses five-fold stratified out-of-fold (OOF) teacher predictions on real training rows.

Synthetic query pools. The generated-to-real ratio is varied over $K \in \{1, 5, 10, 20, 30, 40\}$. For each generator, each student seed produces a fresh pool of Kn_{train} synthetic rows. Generated labels are not used as hard targets in binary Stage 1. In multiclass Stage 1, they only identify the target class in the decoupled distillation loss; no cross-entropy loss is applied to generated labels. Teacher probabilities are obtained by querying the frozen teacher on the generated covariates.

Generator configurations. The four generators use the following settings. TabPFGen uses 300 SGLD steps, balanced class sampling, and a four-estimator label refiner. Copula uses class-conditional Gaussian copulas with empirical class proportions. CTGAN uses 300 training epochs, a batch size of 500, and the natural class distribution. TabDiff uses 500 training steps, a batch size of 256, validation every 50 steps, best-checkpoint selection beginning at step 100, and early-stopping patience 200. All other parameters follow their default settings.

Student architecture and training objectives. The default student is a residual MLP with four LayerNorm–SiLU blocks, hidden width 512, and dropout 0.2. With temperature $T = 2$ and probability clipping at $\varepsilon = 10^{-7}$, binary Stage 1 uses temperature-scaled soft-label cross-entropy. Multiclass Stage 1 uses decoupled knowledge distillation with equal target-class and non-target-class weights and no hard-label term. Stage 2 is warm-started from the best Stage 1 model and combines the soft teacher objective with hard real-label cross-entropy using weights 0.7 and 0.3, respectively.

Optimization and evaluation. All neural students use AdamW, cosine learning-rate annealing, gradient-norm clipping at 1, and validation-AUC early stopping with patience 30. Stage 1 and the supervised MLP baseline use an initial learning rate of 5×10^{-5}, weight decay 10^{-4}, and a 300-epoch cap. Their cosine schedules have $\eta_{\min} = 5 \times 10^{-7}$. Stage 2 uses an initial learning rate of 10^{-5}, weight decay 2×10^{-6}, a 200-epoch cap, and $\eta_{\min} = 10^{-7}$. The effective batch size is $\min(4096, n_{\text{pool}})$ for Stage 1, $\min(512, n_{\text{train}})$ for Stage 2, and $\min(4096, n_{\text{train}})$ for the supervised MLP baseline. We restore the checkpoint with the best validation AUC after each training run. Stage 2 freezes batch-normalization statistics and uses the Stage 1 validation AUC as a non-degradation floor. Unless otherwise stated, all results are averaged over five seeds, $\{42, 43, 44, 45, 46\}$. We report ROC-AUC for binary tasks and macro one-vs-rest AUC for multiclass tasks.

Official TALENT supervised baselines. RealMLP and TabM are evaluated using the released default configurations of TALENT, without dataset-specific hyperparameter tuning. The common TALENT deep-learning protocol uses mean imputation for numerical missing values, a dedicated "new" category for missing categorical values, train-fitted standardization, no numerical binning (`num_policy = none`), and validation-based checkpoint selection. For the RealMLP wrapper, categorical features use the required index representation; TabM receives ordinal category indices through its embedding path. The configuration used for the reported baselines is summarized below.

Table 5: Configuration of the supervised RealMLP and TabM baselines. The values are taken from TALENT's released default model configuration files and global deep-learning configuration, with the paper's five-run seed protocol replacing TALENT's example seed count.

Component	Effective TALENT configuration
Global data and training	`normalization` = `standard`; `num_nan_policy` = `mean`; `cat_nan_policy` = `new`; `num_policy` = `none`; `n_bins` = 2; `cat_min_frequency` = 0; `max_epoch` = 200; `batch_size` = 1024; `evaluate_option` = `best-val`; `workers` = 0.
RealMLP model	`num_emb_type` = `pbld`; `add_front_scale` = `true`; `lr` = 4×10^{-2}; `p_drop` = 0.15; `act` = `selu`; `hidden_sizes` = $(256, 256, 256)$; `wd` = 2×10^{-2}; `plr_sigma` = 0.1; `ls_eps` = 0.1; `cat_policy` = `indices`.
TabM model	`arch_type` = `tabm`; $k = 32$; PLR embeddings with `n_frequencies` = 77, `frequency_scale` = 0.04431360576139521, `d_embedding` = 34, and `lite` = `true`; MLP backbone with `n_blocks` = 2, `d_block` = 384, and `dropout` = 0.1; `lr` = 3×10^{-4}; `weight_decay` = 10^{-5}.
Tuning and randomness	`tune` = `false` and `retune` = `false`; no per-dataset search is performed. For each reported seed $s \in \{42, 43, 44, 45, 46\}$, we follow TALENT's seed utility: Python uses s, NumPy uses $s + 1$, PyTorch uses $s + 2$, and CUDA uses $s + 3$. DataLoader workers are disabled (`workers` = 0); the official CUDA backend behavior is retained.

Table 6: Inference-time summary across TALENT datasets. Entries report the median of the dataset-level mean inference times in milliseconds, with interquartile ranges in parentheses. TFM GPU timings measure inference only on a single NVIDIA A800-SXM GPU; a dash indicates an unmeasured device setting.

Device	TabICL	TabPFN	TabDPT	MLP	XGB	LGB
GPU	593.4 (289.4–1301.5)	785.2 (452.3–1485.5)	351.3 (157.0–1443.3)	1.04 (0.98–1.65)	1.43 (0.85–3.13)	–
CPU	–	–	–	6.13 (3.76–11.78)	1.57 (0.89–5.88)	0.49 (0.17–2.36)

The global settings are kept separate from model-specific settings so that the effective protocol can be reproduced without relying on undocumented library defaults.

Efficiency measurement. Timing experiments use the server described in Section 4.1. Direct TFM inference is measured on an NVIDIA A800-SXM GPU, whereas student inference is measured on the CPU and, when supported, on the same GPU. We report medians and interquartile ranges of dataset-level mean inference times.

E.4 Inference-Time Summary

E.5 Generator and Stage 1 Ablations

Tree-student configuration. The tree-student experiments use the normalized feature space, seed 42, the generated-to-real ratios $K = 30$, temperature $T = 2$, and probability clipping at $\varepsilon = 10^{-7}$. Stage 1 performs pure soft distillation on the generated covariates. Stage 2 warm-starts from the Stage 1 model and fits real training examples with a weighted objective consisting of 0.7 soft distillation and 0.3 hard-label cross-entropy. For multiclass distillation, all three backends use the forward-KL objective; the hard-label baselines use their native binary or multiclass objectives. The Stage 2 S1-anchor guard retains the Stage 1 test predictions whenever the Stage 2 validation AUC is lower than the Stage 1 validation AUC.

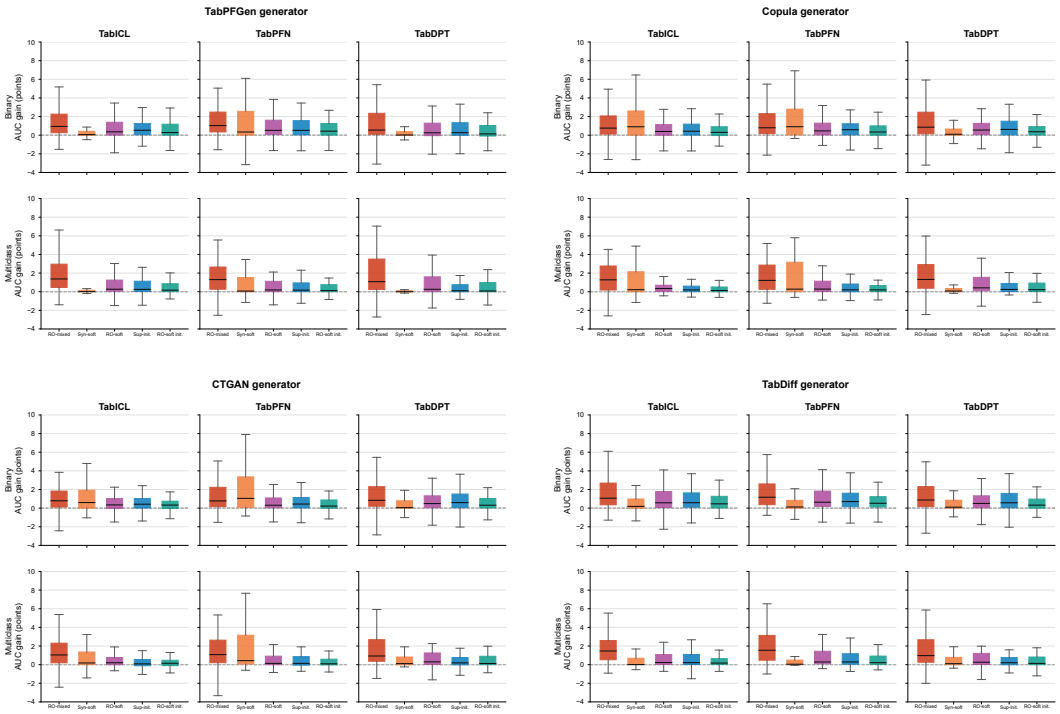


Figure 7: Full Stage 1 replacement ablations for TabPFGen (top left), Copula (top right), CTGAN (bottom left), and TabDiff (bottom right). In each panel, the boxplots are ordered from left to right as follows: real-only mixed distillation, synthetic-only soft distillation, real-only soft distillation, supervised MLP initialization, and real-only soft initialization. Each boxplot reports the paired AUC-point improvement of GEAR relative to the specified alternative.

Table 7: Hyperparameters for the tree-student experiments. The Stage 1/CE rows share the same structural settings; CE denotes the hard-label baseline with the backend's native objective.

Backend	Regime	Hyperparameters
LightGBM	Stage 1 / CE	*num_leaves* = 63, *learning_rate* = 0.1, *feature_fraction* = 0.8, *bagging_fraction* = 0.8, *bagging_freq* = 1, *min_data_in_leaf* = 20, four threads; 500 boosting rounds with 50-round early stopping.
LightGBM	Stage 2	*num_leaves* = 63, *learning_rate* = 0.05, *feature_fraction* = 0.8, *bagging_fraction* = 0.8, *bagging_freq* = 1, *min_data_in_leaf* = 10, four threads; 300 boosting rounds with 30-round early stopping.
XGBoost	Stage 1 / CE	Histogram tree method, *max_depth* = 6, *eta* = 0.1, *subsample* = 0.8, *colsample_bytree* = 0.8, *min_child_weight* = 1, four threads; 500 boosting rounds with 50-round early stopping. The histogram learner uses the GPU when available and otherwise runs on the CPU.
XGBoost	Stage 2	Histogram tree method, *max_depth* = 6, *eta* = 0.05, *subsample* = 0.8, *colsample_bytree* = 0.8, *min_child_weight* = 1, four threads; 300 boosting rounds with 30-round early stopping. The device setting is the same as in Stage 1.
CatBoost		500 iterations, *learning_rate* = 0.1, *depth* = 6, *l2_leaf_reg* = 3.0, *rsm* = 0.8, Bayesian bootstrap, four threads, and 50-round early stopping. Intermediate files are disabled.

Table 8: Complete tree-student ablation across generators and teachers. Entries report mean AUC (standard error) across paired datasets. For each student row, the Stage 1 and Stage 2 entries use the synthetic-volume setting that maximizes the Stage 2 mean AUC. Boldface indicates the best result among Stage 1, real-only distillation, and Stage 2 for each student.

Generator	Teacher	Task	LightGBM			XGBoost		
			Stage 1	Real-only	Stage 2	Stage 1	Real-only	Stage 2
TabPFGen	TabICL	Binary	0.855 (0.009)	0.855 (0.009)	**0.861** (0.009)	0.858 (0.009)	0.860 (0.009)	**0.862** (0.009)
		Multiclass	0.863 (0.014)	0.841 (0.015)	**0.867** (0.014)	0.923 (0.011)	0.925 (0.011)	**0.925** (0.011)
	TabPFN	Binary	0.820 (0.011)	0.855 (0.009)	**0.860** (0.009)	0.819 (0.012)	0.860 (0.009)	**0.862** (0.009)
		Multiclass	0.819 (0.017)	0.807 (0.015)	**0.864** (0.013)	0.864 (0.017)	0.921 (0.012)	**0.924** (0.011)
	TabDPT	Binary	0.854 (0.009)	0.853 (0.009)	**0.858** (0.009)	0.856 (0.009)	0.857 (0.009)	**0.858** (0.009)
		Multiclass	0.885 (0.012)	**0.895** (0.012)	0.892 (0.012)	0.918 (0.011)	0.917 (0.012)	**0.920** (0.011)
Copula	TabICL	Binary	0.831 (0.010)	0.855 (0.009)	**0.861** (0.009)	0.833 (0.010)	0.860 (0.009)	**0.861** (0.009)
		Multiclass	0.841 (0.014)	0.814 (0.015)	**0.860** (0.013)	0.903 (0.013)	0.925 (0.011)	**0.926** (0.011)
	TabPFN	Binary	0.820 (0.011)	0.855 (0.009)	**0.862** (0.009)	0.819 (0.011)	0.860 (0.009)	**0.862** (0.009)
		Multiclass	0.808 (0.017)	0.807 (0.015)	**0.853** (0.014)	0.870 (0.017)	0.921 (0.012)	**0.924** (0.012)
	TabDPT	Binary	0.839 (0.009)	0.853 (0.009)	**0.860** (0.009)	0.843 (0.009)	0.858 (0.009)	**0.859** (0.009)
		Multiclass	0.876 (0.013)	0.895 (0.012)	**0.896** (0.012)	0.901 (0.013)	0.917 (0.012)	**0.921** (0.011)
CTGAN	TabICL	Binary	0.832 (0.010)	0.854 (0.009)	**0.859** (0.009)	0.821 (0.010)	**0.859** (0.009)	0.857 (0.009)
		Multiclass	0.835 (0.015)	0.811 (0.015)	**0.853** (0.015)	0.905 (0.012)	0.923 (0.012)	**0.924** (0.011)
	TabPFN	Binary	0.803 (0.011)	0.855 (0.009)	**0.861** (0.009)	0.806 (0.011)	0.859 (0.009)	**0.860** (0.009)
		Multiclass	0.792 (0.019)	0.807 (0.015)	**0.842** (0.015)	0.851 (0.018)	0.921 (0.012)	**0.924** (0.011)
	TabDPT	Binary	0.838 (0.010)	0.853 (0.009)	**0.858** (0.009)	0.838 (0.010)	0.857 (0.009)	**0.858** (0.009)
		Multiclass	0.863 (0.013)	0.895 (0.012)	**0.898** (0.012)	0.895 (0.013)	0.917 (0.012)	**0.921** (0.011)
TabDiff	TabICL	Binary	0.844 (0.010)	0.855 (0.009)	**0.861** (0.009)	0.846 (0.010)	0.860 (0.009)	**0.860** (0.009)
		Multiclass	0.869 (0.013)	0.814 (0.015)	**0.879** (0.013)	0.918 (0.012)	0.925 (0.011)	**0.927** (0.011)
	TabPFN	Binary	0.848 (0.010)	0.855 (0.009)	**0.862** (0.009)	0.849 (0.009)	0.860 (0.009)	**0.862** (0.009)
		Multiclass	0.857 (0.013)	0.807 (0.015)	**0.868** (0.012)	0.912 (0.012)	0.921 (0.012)	**0.925** (0.011)
	TabDPT	Binary	0.840 (0.010)	0.853 (0.009)	**0.860** (0.009)	0.840 (0.010)	0.858 (0.009)	**0.859** (0.009)
		Multiclass	0.870 (0.013)	0.895 (0.012)	**0.898** (0.012)	0.896 (0.013)	0.917 (0.012)	**0.921** (0.011)

Table 9: Complete plot-level statistics underlying Figure 1, grouped by teacher. The MLP rows include the four-block default and one-to-three-block ablations; the no-ResNet row is reported for TabICL only. For each TFM teacher row, the daggered AUC column reports teacher mean AUC; for student and baseline rows, it reports supervised AUC. Real-only and GEAR AUC are reported for the available distilled students, whereas CatBoost and RealMLP are supervised-only baselines. Time and peak RSS are reported in milliseconds and gigabytes, respectively; the symbol "–" denotes an unavailable measurement.

Model	Supervised AUC†	Real-only AUC	GEAR AUC	Time (ms)	RSS (GB)
Teacher: TabPFN					
TFM teacher	0.8846	–	–	772.626	2.146
MLP (1 block)	0.8506	0.8306	0.8652	1.147	0.659
MLP (2 blocks)	0.8508	0.8394	0.8661	1.717	0.656
MLP (3 blocks)	0.8501	0.8440	0.8656	2.032	0.659
MLP (4 blocks)	0.8498	0.8459	0.8677	6.266	0.670
LGB	0.8510	0.8643	0.8660	0.270	0.644
XGB	0.8534	0.8652	0.8675	0.968	0.667
TabM	0.8488	0.7557	0.8529	3.983	1.285
CatBoost	0.8568	–	–	1.691	1.095
RealMLP	0.8429	–	–	36.212	1.627
Teacher: TabICL					
TFM teacher	0.8847	–	–	558.135	1.424
MLP (1 block)	0.8506	0.8310	0.8674	1.144	0.672
MLP (2 blocks)	0.8508	0.8398	0.8673	1.719	0.659
MLP (3 blocks)	0.8501	0.8442	0.8675	2.042	0.669
MLP (4 blocks)	0.8498	0.8452	0.8672	6.256	0.670
MLP (4 blocks no ResNet)	0.8505	0.8317	0.8667	6.265	0.651
LGB	0.8510	0.8649	0.8680	0.272	0.646
XGB	0.8534	0.8651	0.8679	0.977	0.635
TabM	0.8488	0.7557	0.8641	3.981	1.290
CatBoost	0.8568	–	–	1.631	1.080
RealMLP	0.8429	–	–	36.410	1.633
Teacher: TabDPT					
TFM teacher	0.8820	–	–	355.274	1.249
MLP (1 block)	0.8506	0.8325	0.8641	1.143	0.659
MLP (2 blocks)	0.8508	0.8408	0.8643	1.719	0.656
MLP (3 blocks)	0.8501	0.8458	0.8640	2.137	0.658
MLP (4 blocks)	0.8498	0.8478	0.8648	6.163	0.668
LGB	0.8510	0.8631	0.8646	0.275	0.650
XGB	0.8534	0.8646	0.8640	0.938	0.663
TabM	0.8488	0.7564	0.8627	3.977	1.279
CatBoost	0.8568	–	–	1.654	1.091
RealMLP	0.8429	–	–	36.290	1.629

E.6 Complete Tree-Student Results

E.7 Numerical Data for Figure 1

Table 9 reports the values underlying Figure 1; the TabM real-only results are included for completeness. Mean AUC is averaged over matched TALENT binary tasks, whereas inference time and peak resident set size (RSS) are medians across datasets. TFM rows report peak GPU memory on an A800-SXM GPU; student and baseline rows use their corresponding deployment protocols.