RECAST: RESIDUAL-STATE REPRESENTATION LEARNING WITH FOUNDATION MODELS FOR LIGHTWEIGHT TABULAR PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Even well-trained tabular models can exhibit systematic error patterns in particular regions of the feature space. We propose RECAST, which learns residual-state representations to improve an existing model, rather than directly relearning its residuals from the original features alone. At each correction stage, RECAST partitions out-of-fold XGBoost errors into three states. An LLM uses feature semantics and aggregated error statistics to select feature subsets and propose a focus condition that guides in-context example selection, while a tabular foundation model (TFM) estimates residual-state representations for new inputs. A teacher augments the original features with these representations to predict residual corrections, and the residual states and context are updated after each stage. The teacher's out-of-fold predictions are then blended with ground-truth labels to train a single XGBoost student using only the original features. Foundation models are used only during training; inference requires only the student and the original inputs. Across nine regression and nine binary classification datasets from OpenML, RECAST achieves the best average rank among ten compared models: 4.11 for regression, 3.00 for classification, and 3.56 overall. It improves the baseline XGBoost's three-seed mean performance on all 18 datasets. Controlled comparisons further show better task-level mean performance than residual relearning without state representations and direct distillation of the TFM's original-target predictions. These results support residual-state estimation as a training-time use of foundation models for improving an existing tabular predictor while retaining lightweight inference.

1 INTRODUCTION

Gradient-boosted decision trees (GBDTs) remain strong baselines for tabular data (Friedman, 2001; Chen & Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018; Grinsztajn et al., 2022). Recent neural models such as RealMLP and TabM also perform strongly (Holzmüller et al., 2024; Gorishniy et al., 2024), while tabular foundation models (TFMs), including TabPFN and LiMiX, offer pretrained in-context prediction (Hollmann et al., 2025; Zhang et al., 2025b). Rather than replacing an existing predictor with another model family, we ask: *Can an already well-trained model be improved by exploiting the structure of the errors it leaves behind?*

Directly fitting residuals and learning a representation for residual correction address different questions. The former asks whether another model can fit the errors from the same inputs; the latter asks whether those errors can be expressed in a form that a restricted correction model can use more effectively. Our aim is not to obtain additional observed variables at prediction time, but to construct an auxiliary representation from the inputs already available. Residual-state estimation provides this representation, and its contribution is assessed against the same correction-and-distillation procedure without state representations.

We propose RECAST, which turns residual-state estimation into an auxiliary task for prediction correction. Residuals are partitioned into a small number of states, and their probabilities for new inputs serve as soft representations. A correction model receives these probabilities alongside the original features and directly predicts residual values. The distinction from residual relearning is therefore

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.

the representation supplied to the correction model, not merely an additional training stage. These states characterize errors observed on training samples, whereas their representations for new inputs must be estimated from features alone. The auxiliary task thus connects the baseline's observed failures to an input-dependent description of how it is likely to err.

The foundation models have complementary roles. The LLM uses feature semantics and residual-state statistics to select features and examples; the TFM uses the resulting context to estimate input-specific state probabilities rather than the original target. An XGBoost teacher corrects predictions using these representations, then refreshes the states and context from the remaining residuals. Finally, a single XGBoost student learns from targets blending the teacher's out-of-fold (OOF) predictions with ground-truth labels. Distillation transfers the corrected teacher's predictive information, not the foundation models' outputs directly. The student uses only the original features, removing the LLM, TFM, and multistage teacher from inference.

Our contributions are threefold.

1. *Residual-state representations for correction.* We construct auxiliary inputs that characterize the current model's errors and evaluate their benefit beyond residual relearning without these representations.
2. *Complementary LLM and TFM roles.* LLM-guided context construction and input-specific state estimation are repeated on the residuals remaining after correction.
3. *Transfer to a lightweight student.* Across 18 OpenML datasets, the final XGBoost student achieves the best average rank among the compared models in both tasks and improves the baseline's three-seed mean performance on every dataset, with comparable inference cost.

2 Related Work

Tabular learning and residual correction. Gradient boosting sequentially addresses a model's remaining errors (Friedman, 2001), with XGBoost, LightGBM, and CatBoost as prominent implementations (Chen & Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018). Tree models remain competitive alongside neural approaches (Grinsztajn et al., 2022; Holzmüller et al., 2024; Gorishniy et al., 2024). RECAST likewise learns residuals, but changes the correction inputs through estimated residual-state representations. Its key comparison retains correction, distillation, and model selection while removing those representations, rather than omitting further training altogether. This asks whether an error-focused representation provides an advantage beyond fitting another residual model in the original feature space.

LLM-assisted tabular learning. LIFT and TabLLM use language interfaces or verbalized rows for prediction (Dinh et al., 2022; Hegselmann et al., 2023). Related directions include TABLET, UniPredict, pretrained prediction, and transfer learning (Slack & Singh, 2023; Wang et al., 2023; Yan et al., 2024; Gardner et al., 2024), including medical and few-shot settings (Wang et al., 2024; Kang et al., 2026; Wang et al., 2026; Chen et al., 2026). CAAFE and FeatLLM generate features (Hollmann et al., 2023; Han et al., 2024); LLM-Select and LLM-Lasso use semantics for selection or regularization (Jeong et al., 2025; Zhang et al., 2025a). Other methods derive priors or search for features (Choi et al., 2022; Zhu et al., 2023; Zhang et al., 2026; Shi et al., 2026; Abhyankar et al., 2026). These uses target the original prediction problem; RECAST instead uses semantics to organize information about an existing model's observed errors.

LLMs and tree models. OCTree generates features through decision-tree reasoning (Nam et al., 2024); DeLTa combines LLMs and decision trees (Ye et al., 2025); Talking Trees studies reasoning-based tree induction (Yakushev et al., 2025). RECAST instead derives auxiliary supervision from actual OOF errors. LLM proposals construct context for distinguishing residual states, and the TFM converts this context into probabilities for teacher correction. Neither LLM-generated rules nor state probabilities enter the deployed student; their predictive effect is transferred through the corrected teacher. The supervision, training-time use of foundation models, and deployment path therefore distinguish the method, rather than the use of an LLM or tree model alone.

TFMs and distillation. TabPFN and LiMiX provide pretrained in-context prediction (Hollmann et al., 2025; Zhang et al., 2025b), while TabSTAR and ConTextTab incorporate text or semantic

information (Arazi et al., 2025; Spinaci et al., 2025). RECAST supplies residual-state labels to the TFM and uses its probabilities as correction inputs. Distillation transfers information to simpler models (Hinton et al., 2015; Frosst & Hinton, 2017; Yan et al., 2025), with cross-fitting and theoretical analyses studied in prior work (Dao et al., 2021; Boix-Adserà, 2024). Unlike direct TFM distillation, RECAST blends complete residual-corrected teacher predictions with labels, separating training-time representation estimation from the original-feature-only student. The direct-TFM control tests this distinction by assigning the TFM the original target instead of residual states.

3 METHOD

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathcal{X}$ contains the original features available at prediction time. We denote the base model by F_0, the teacher at correction stage $h \in \{1, \ldots, H\}$ by T_h, and the student by S. Training, validation, and independent test data are $\mathcal{D}_{\mathrm{tr}}$, $\mathcal{D}_{\mathrm{val}}$, and $\mathcal{D}_{\mathrm{test}}$. Figure 1 summarizes the pipeline.

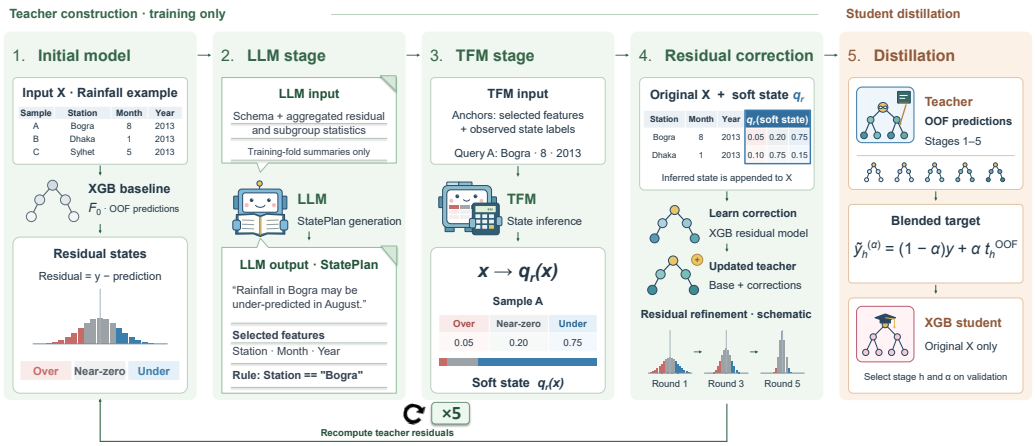


Figure 1: RECAST training pipeline. OOF errors define residual states; LLM-guided context and TFM estimation provide representations for teacher correction. States and context are refreshed after correction. Blended OOF teacher targets train an original-feature-only student. Feature names, state names, and probabilities are illustrative.

3.1 DEFINING RESIDUAL STATES

Let G_{h-1} denote the predictor entering stage h, with $G_0 = F_0$, $G_{h-1} = T_{h-1}$ $(h \geq 2)$. For regression, define the residuals and state labels as

$$r_i^{(h)} = y_i - G_{h-1}^{\mathrm{OOF}}(x_i),\ C_i^{(h)} = b(r_i^{(h)}) \in \{1, \ldots, K\}.$$

Here, $G_{h-1}^{\mathrm{OOF}}(x_i)$ denotes the out-of-fold prediction associated with training row i within the current teacher-fitting context. The partition $b(\cdot)$ is determined from the permitted training data and then fixed. Negative residuals indicate overprediction and positive residuals underprediction. We apply deterministic one-dimensional k-means to signed residuals to obtain three states; they need not correspond one-to-one to overprediction, near-zero errors, and underprediction. For binary classification, fixed-margin binning partitions $m = p(y) - p(1-y)$ into $m < -\delta$, $|m| \leq \delta$, and $m > \delta$, with $\delta = 0.2$. Here, $p(y)$ is the probability assigned to the true class. Appendix A gives task-specific settings.

The observed residual-state label $C_i^{(h)}$ is used to supervise the estimation of $q_r(x) \in \Delta^{K-1}$ from the input features, where $\Delta^{K-1} = \{\boldsymbol{u} \in \mathbb{R}_{\geq 0}^{K} : \sum_{k=1}^{K} u_k = 1\}$. We write $q_r^{(h)}(x)$ when emphasizing the stage. The correction model receives this probability vector, not a single state label, and is never given a new sample's ground truth or actual residual.

3.2 LLM–TFM RESIDUAL-STATE ESTIMATION

Feature selection and context construction. The LLM receives the task description, schema, and aggregated error and state statistics, including feature summaries and high-error subgroup statistics rather than raw individual rows. It proposes features, a focus rule for sample selection, and an explanatory hypothesis:

$$\mathcal{H}_h = \mathcal{A}_{\mathrm{LLM}}(\mathrm{schema}, \mathrm{task}, \mathrm{state\ summaries})$$

The proposals guide feature and sample selection from permitted training data; they do not generate state labels or final predictions. Let I_h contain the selected sample indices and ϕ_h apply feature selection and fixed preprocessing. The in-context learning (ICL) context is

$$\mathcal{C}_h = \{(\phi_h(x_i), C_i^{(h)}) : i \in I_h\}.$$

Context labels come from actual residuals. We verify feature existence, exclude targets, identifiers, and features generated after prediction time. We also check for duplicates and verify context size. The LLM selects up to eight features. Each context contains at most 256 rows, with at least one anchor from each nonempty state. The remaining rows are sampled in proportion to the observed state frequencies. A focus condition prioritizes anchors rather than assigning labels. The selected feature names must correspond to observed inputs, and the query's original target is never supplied through the TFM input channel. Appendix A specifies sampling and provides a recorded prompt–response example.

State representation estimation with a TFM. Given the stage-h context, the TFM Q_θ estimates the residual-state representation of an input x as

$$\hat{q}_r^{(h)}(x) = Q_\theta(\phi_h(x); \mathcal{C}_h) \in \Delta^{K-1}.$$

This probability vector is the residual-state representation. Using a probability distribution rather than a hard state assignment allows the correction model to account for uncertainty in state membership. The LLM determines which information and examples are provided, while the TFM estimates the input-specific states from that context. Even in regression, the auxiliary task distinguishes states rather than predicting continuous residuals. A query supplies only its transformed input $\phi_h(x)$.

For a fixed estimator, $q_r(X)$ adds no new observed information, but it can change how a restricted correction model represents residual structure. A residual pattern requiring complex partitions of X may admit simpler splits on $q_r(X)$ under the same tree constraints. Appendix F analyzes this potential approximation benefit without guaranteeing gains for every representation.

3.3 RESIDUAL CORRECTION AND ITERATIVE LEARNING

The regression correction model predicts residuals from the original features and cross-fitted state representations. Let $\tilde{q}_{r,i}^{(h)}$ denote the cross-fitted residual-state estimate for training row i at stage h. We learn the correction model as

$$\hat{d}_h = \operatorname*{arg\,min}_{d \in \mathcal{G}_T} \frac{1}{n} \sum_i \big(r_i^{(h)} - d(x_i, \tilde{q}_{r,i}^{(h)})\big)^2 + \lambda_T \Omega_T(d), \tag{1}$$

$$T_h(x) = G_{h-1}(x) + \gamma \hat{d}_h\big(x, \hat{q}_r^{(h)}(x)\big). \tag{2}$$

Here, $\hat{d}_h$ is the learned correction and γ controls its magnitude; we fix $\gamma = 0.25$. The training-row representation $\tilde{q}_{r,i}^{(h)}$ is distinguished from $\hat{q}_r^{(h)}(x)$, the state representation computed when evaluating the teacher on a new input. For classification, the correction inputs for training row i additionally include the current cross-fitted positive-class probability $\hat{y}_{\mathrm{prev},i}$, giving $[x_i, \hat{y}_{\mathrm{prev,i}}, \tilde{q}_{r,i}^{(h)}]$. The control without state representations receives x_i for regression and $[x_i, \hat{y}_{\mathrm{prev},i}]$ for classification. For binary classification, the correction target is $y_i - \hat{y}_{\mathrm{prev},i}$.

After each stage, OOF predictions of T_h update the residuals, states, and context, targeting errors left by earlier corrections. Rather than keeping the initial residual partition fixed, each stage uses the corrected model's remaining errors to construct its next auxiliary task. We construct candidates for up to five stages. Classification corrects probabilities and clips them to $[10^{-6}, 1 - 10^{-6}]$ after each stage. The correction model and student reuse the XGBoost hyperparameters and boosting rounds selected for F_0 in the corresponding fitting context. Algorithm 1 in Appendix A gives the full procedure.

3.4 DISTILLATION INTO A SINGLE XGBOOST STUDENT

For each stage, define $\mathbf{t}_h^{\mathrm{OOF}} = \left(T_h^{(-l(1))}(x_1), \ldots, T_h^{(-l(n))}(x_n)\right)^\top$, where $(-l(i))$ excludes sample i's outer fold from the teacher's entire training procedure. At the selected stage $\hat{h}$, let $t_i^{\mathrm{OOF}} = T_{\hat{h}}^{(-l(i))}(x_i) = (\mathbf{t}_{\hat{h}}^{\mathrm{OOF}})_i$. The regression student learns from blended targets:

$$\tilde{y}_i^{(\alpha)} = (1-\alpha)y_i + \alpha t_i^{\mathrm{OOF}},\ \alpha \in [0,1],$$
$$S_\alpha = \arg\min_{s \in \mathcal{G}_S} \frac{1}{n} \sum_i \left(\tilde{y}_i^{(\alpha)} - s(x_i)\right)^2 + \lambda_S \Omega_S(s)$$

At $\alpha = 0$, the student is trained only on ground-truth labels; at $\alpha = 1$, it is trained only on teacher predictions. Intermediate values blend the two. Because the targets contain the teacher's complete corrected predictions, the final predictor is $S_{\hat{\alpha}}(x)$, without adding $F_0(x)$ again. Thus, the student uses only the original features and receives the teacher's information through the blended targets. Consequently, the model deployed for prediction does not reproduce the teacher's sequence of corrections or require state estimation for a new sample.

For binary classification, let $p_{T,i}^{\mathrm{OOF}} = T_{\hat{h}}^{(-l(i))}(x_i)$ be the teacher's positive-class probability. We use $\tilde{p}_i = (1-\alpha)y_i + \alpha p_{T,i}^{\mathrm{OOF}}$. All evaluated classification tasks are binary; Appendix A describes the multiclass extension.

Cross-fitting and model selection. Because residual states depend on labels, cross-fitting covers the entire teacher pipeline. Within each outer-training fold, we perform base-model HPO and fitting, preprocessing, inner cross-fitting, state definition, LLM planning, context construction, and all correction stages. Each teacher predicts its held-out outer fold; the predictions are assembled into OOF student targets. Held-out labels are excluded from the current and all preceding stages, and a query's actual state is not included in its own context (Dao et al., 2021).

Students are trained for $h \in \{1, 2, 3, 4, 5\}$ and $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. We jointly select the stage and weight by student validation RMSE for regression or AUROC for classification, with ties favoring an earlier stage and then a smaller α. The independent test set is used only after selection (Cawley & Talbot, 2010). Only the original-feature-only student is deployed.

Theoretical interpretation. Appendix F analyzes regression under squared risk. Beyond the representation argument above, it derives the exact risk difference from residual correction and characterizes the population objective of OOF blended targets as an interpolation between the conditional mean and a weighted average of fold-specific teachers. This population objective is distinct from simply imitating a teacher. These results do not automatically guarantee finite-sample gains for the learned teacher or student; final-task performance is evaluated empirically below.

4 EXPERIMENTS

Evaluation setup. We evaluate nine regression and nine binary classification datasets from OpenML, covering small- and medium-scale tabular problems. The ten models are RECAST, XGB, LGBM, CatBoost, TabM, RealMLP, TabPFN3, LiMiX-2M, OCTree, and DeLTa. Our default uses Qwen3.5-122B-A10B-GPTQ-Int4 and LiMiX-2M. Each base XGBoost fitting context searches 32 hyperparameter candidates. Three seeds define 60/20/20 training/validation/test splits, with outer five-fold and inner three-fold cross-fitting. For regression, we assign rows with identical feature values to the same split; for classification, we use class-stratified splits. Preprocessing is fitted only on the training rows available in each fitting context. Appendix A gives preprocessing, model settings, and tuning procedures. Equal HPO candidate counts do not imply equal total training cost or validation use; RECAST additionally selects the stage and mixing weight.

Regression uses NRMSE and classification uses AUROC. Models are ranked by three-seed mean performance on each dataset; ranks are averaged equally across datasets, rounding only the final aggregate. Average rank is the primary aggregate metric. Appendix E defines the metrics and standard deviations, and Appendix B reports complete results.

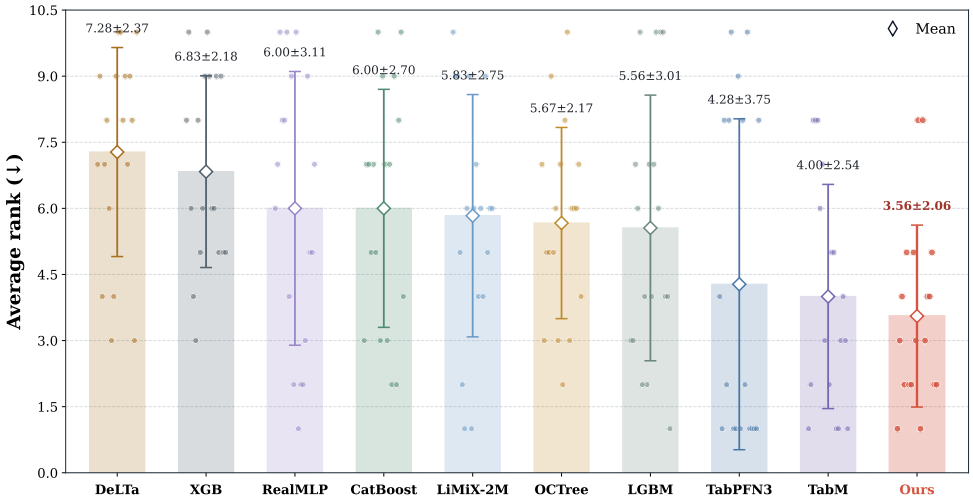


Figure 2: Rank distributions across 18 datasets. Bars and diamonds show mean ranks; error bars show standard deviations across datasets. Dots are per-dataset ranks. Lower is better.

4.1 Student Performance and Per-Dataset Improvements

The RECAST student achieves the best average rank among the ten models in both tasks: 4.11 for regression and 3.00 for classification (Table 10, Appendix B). Its overall rank is 3.56, versus 4.00 for TabM and 4.28 for TabPFN3. Figure 2 shows rank standard deviations of 2.06, 2.54, and 3.75, respectively. These are descriptive statistics across datasets, not variation across training seeds.

Table 1: Base XGB versus RECAST student: three-seed mean performance. Decrease and increase are relative NRMSE reductions and AUROC gains (%), respectively. Bold marks the better mean.

(a) Regression: NRMSE ↓

Dataset	XGB	Ours	Decrease (%)
Diabetes	0.8549	**0.8155**	4.61
Forest	2.4792	**2.4666**	0.51
Puma8NH	0.5783	**0.5769**	0.24
Rainfall	0.4832	**0.4823**	0.19
Optical	0.3477	**0.3459**	0.52
Vineyard	0.4039	**0.3916**	3.05
Strikes	1.2142	**1.1727**	3.42
Apnea1	0.3185	**0.2893**	9.17
Places	0.8794	**0.8679**	1.31
Improved / Tied / Worse: 9 / 0 / 0			

(b) Classification: AUROC ↑

Dataset	XGB	Ours	Increase (%)
Adult	0.9308	**0.9312**	0.04
Mammography	0.9437	**0.9528**	0.96
Telco-Churn	0.8471	**0.8484**	0.15
Solar-Flare	0.7265	**0.7426**	2.22
PBC	0.7371	**0.7628**	3.49
Eucalyptus	0.8230	**0.8491**	3.17
jEdit	0.6364	**0.6918**	8.71
pc1_req	0.5745	**0.5767**	0.38
art_daily	0.7135	**0.7163**	0.39
Improved / Tied / Worse: 9 / 0 / 0			

Table 1 compares the final student with the baseline XGBoost model using performance. Three-seed mean performance improves on all 18 datasets: mean NRMSE falls from 0.8399 to 0.8232 (1.99%), and mean AUROC rises from 0.7703 to 0.7857 (2.00%). These aggregate gains are not driven by improvements on a few datasets offsetting deterioration on others. Improvement magnitudes vary: NRMSE falls by 9.17% on Apnea1 and 4.61% on Diabetes, while AUROC rises by 8.71% on jEdit. The claim concerns each dataset's three-seed mean, not equal-sized gains or improvement in every seed. Together, average rank and the baseline comparison measure competitiveness among model families and improvement of the model from which the procedure starts.

4.2 Effect of Residual-State Representations

Table 2 separates additional residual learning, TFM use, context construction, and blended distillation. The X-only residual teacher retains iterative correction, distillation, and stage/α selection, removing only state representations; classification retains $\hat{y}_{\mathrm{prev}}$ in both conditions. No LLM retains three states and TFM estimation but randomly selects up to eight features without a focus condition. Direct TFM $\rightarrow$ student predicts the original target using the same outer five-fold split and each fold's training context, then blends its OOF predictions with labels. It selects among five mixing weights without iterative stages. Appendix A gives the detailed controls.

Table 2: Controlled experiments: mean NRMSE over nine regression datasets and mean AUROC over nine classification datasets. Bold marks the best mean. All conditions deploy an XGBoost student using only the original X.

Condition	Mean NRMSE ↓	Mean AUROC ↑
Full method	**0.8232**	**0.7857**
No LLM	0.8364	0.7798
X-only residual teacher $\rightarrow$ student	0.8297	0.7798
Direct TFM $\rightarrow$ student	0.8287	0.7811
$\alpha = 1$ student	0.8674	0.7802

Beyond additional residual learning. Because the X-only control retains the same iterative correction, distillation, and stage/α selection procedure, the Full-versus-X-only comparison isolates the contribution of adding residual-state representations. The X-only residual teacher $\rightarrow$ student improves on base XGBoost, showing that correction and distillation themselves contribute. Adding residual-state representations further reduces mean NRMSE by 0.78% and increases mean AUROC by 0.76%. The full gain therefore cannot be attributed to additional residual training alone. The comparison supports estimating how the current predictor is likely to err and providing this representation as an additional correction input.

Residual-focused TFM use. Relative to direct TFM-to-student distillation, RECAST reduces mean NRMSE by 0.66% and increases mean AUROC by 0.59%. This supports assigning the TFM residual-state estimation for correction rather than directly distilling its original-target predictions. Both approaches use a TFM and deploy an XGBoost student, but the information extracted from the TFM and the route by which it reaches the student differ.

Context construction. Relative to no LLM, the full method reduces mean NRMSE by 1.58% and increases mean AUROC by 0.0059. In regression, no LLM yields 0.8364 NRMSE, worse than X-only residual relearning at 0.8297. Thus, the usefulness of state representations depends on how their context is constructed. This comparison evaluates LLM-guided context construction as a whole, because it changes the feature-selection and anchor-selection procedures without matching feature counts to the LLM plan. It does not isolate semantic information alone from these context-construction choices.

Blending versus pure imitation. Compared with $\alpha = 1$, blended distillation reduces mean NRMSE by 5.10% and increases mean AUROC by 0.70%. Pure imitation yields regression NRMSE of 0.8674, worse than base XGBoost's 0.8399, whereas the full method reaches 0.8232. Improving the teacher and transferring its information are distinct objectives; retaining ground-truth supervision helps translate representation-based learning into student performance. In this comparison, the best student is not obtained by treating the teacher's predictions as the sole training targets.

4.3 Roles of the LLM and State Estimator

Table 3 tests whether gains persist under component replacement. It reports final student performance, not LLM target prediction or state-classification accuracy. Appendix C gives per-dataset results.

Table 3: Final student performance under component replacement. (a) LLM replacement on nine regression datasets; (b) state-estimator replacement on both tasks. Δ is the replacement metric minus the default metric.

(a) LLM replacement: nine regression datasets

LLM (short name)	Mean NRMSE ↓	Δ NRMSE
Qwen3.5-122B (default)	**0.8232**	
Qwen3.5-35B	0.8260	+0.0028
Qwen3.5-9B	0.8303	+0.0071
Qwen2.5-72B	0.8283	+0.0051

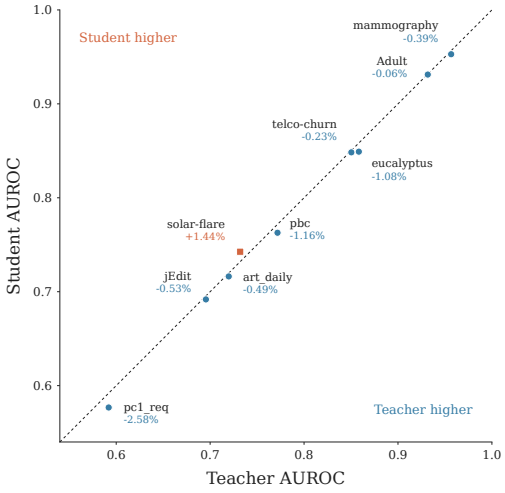
(b) State estimation model replacement

State estimator	NRMSE ↓	Δ NRMSE	AUROC ↑	Δ AUROC
LiMiX-2M (default)	**0.8232**		**0.7857**	
XGBoost	0.8290	+0.0058	0.7814	−0.0043
Random forest	0.8322	+0.0090	0.7839	−0.0018
Logistic regression	0.8305	+0.0073	0.7823	−0.0034

LLM replacement. Qwen3.5-122B has the lowest mean NRMSE, but all three replacements achieve 0.8260–0.8303, improving on base XGBoost's 0.8399. Gains are therefore not restricted to the default LLM. Qwen3.5-35B also outperforms Qwen2.5-72B (0.8260 versus 0.8283), so model size alone does not explain this ordering. These results evaluate the LLM as a context-construction module through the usefulness of its proposals for the final student.

State-estimator replacement. XGBoost, random forest, and logistic regression all improve mean performance over base XGBoost in both tasks. LiMiX-2M performs best, reducing mean NRMSE by 0.70% and increasing mean AUROC by 0.55% relative to the XGBoost estimator. The results distinguish the benefit of state-based correction across estimators from the additional benefit of a TFM. They support residual-state estimation as the organizing auxiliary task, rather than attributing the observed gain only to the default foundation-model checkpoint.

4.4 Teacher–Student Performance and Inference Efficiency



(a) Teacher–student AUROC

Dataset	Teacher AUROC	Student AUROC	Selected α	Selected stage
Adult	**0.9317**	0.9312	0/0.25/0.25	1/4/3
Mammography	**0.9566**	0.9528	0.75/0.25/0.25	5/4/4
Telco-Churn	**0.8504**	0.8484	0/1/0.25	1/2/4
Solar-Flare	0.7320	**0.7426**	1/1/0	4/1/1
PBC	**0.7717**	0.7628	0.5/0.75/1	4/2/3
Eucalyptus	**0.8583**	0.8491	0.75/0.75/0.5	5/4/3
jEdit	**0.6955**	0.6918	1/0.25/1	4/1/1
pc1_req	**0.5920**	0.5767	0/0.25/0.5	1/1/1
art_daily	**0.7198**	0.7163	0/0.5/0	1/5/1

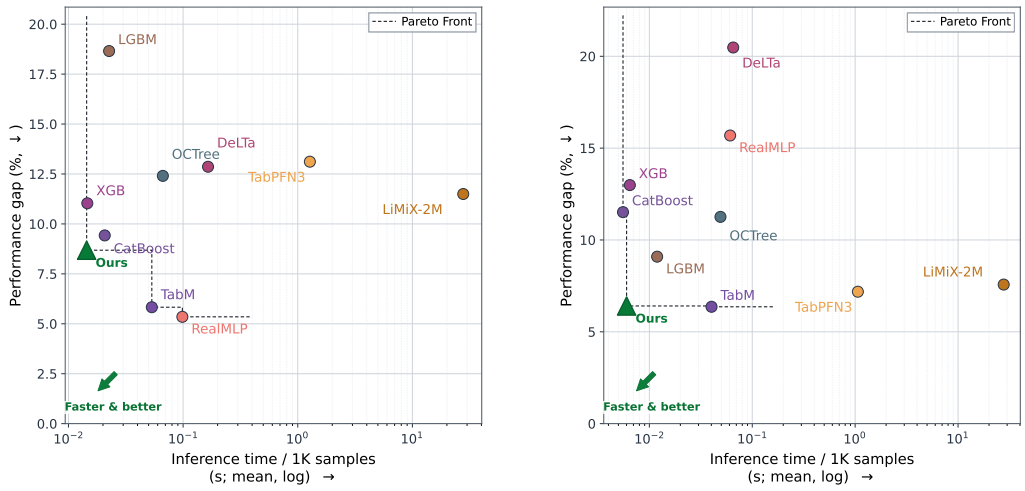
(b) Per-dataset results and selections

Figure 3: Teacher–student test AUROC at the student-selected stage. (a) Dashed line: equal AUROC; annotations: relative changes in mean AUROC. (b) AUROC is averaged over seeds 2032/2033/2034; α and stage follow that order. Bold marks the higher mean.

Teacher–student classification performance. Figure 3 evaluates the teacher at the stage selected by student validation AUROC, not its separately optimized stage. At that stage, final positive-class probabilities from the five fold-specific teachers are averaged per test sample and evaluated on the student's independent test set. This is final-task AUROC, not residual or state-prediction accuracy.

The teacher has higher mean AUROC on eight of nine datasets; on Solar-Flare, the student achieves 0.7426 versus 0.7320. Nevertheless, the student improves on base XGBoost across all nine classification datasets. Selected stages and weights vary across datasets and seeds, including $\alpha = 0$, intermediate values, and $\alpha = 1$. The weight applies only to student targets: at $\alpha = 0$, the teacher score is a reference whose predictions did not train that student.

Inference efficiency. The student lies on the measured Pareto frontier for both tasks (Figure 4) (Deb, 2001). Performance gap measures the relative error reduction needed to reach the best observed performance, using NRMSE for regression and $1 - \mathrm{AUROC}$ for classification; Appendix E gives its definition and aggregation.



(a) Regression

(b) Classification

Figure 4: Inference time versus performance. Time is seconds per 1,000 samples (log scale); performance gap is the mean relative error gap to the best observed model (lower is better; Appendix E). Triangles denote RECAST; dashed lines mark Pareto frontiers.

Separate measurements on three representative datasets and three seeds (Table 17, Appendix D) give student latencies of 0.0144 ms/sample for regression and 0.0060 ms/sample for classification, comparable to XGBoost's 0.0146 and 0.0065 ms/sample. These measurements indicate comparable inference cost; the main benefit is improved predictive performance. Here, lightweight refers to the deployed model and its inference cost, not equal total training cost.

5 Conclusion

RECAST uses residual-state estimation as an auxiliary task for prediction correction and transfers the corrected teacher's information to a single XGBoost student. Across 18 OpenML datasets, the student achieves the best average rank among the compared models in regression and classification and improves the baseline's three-seed mean performance on every dataset. Controls support benefits beyond residual relearning alone and direct TFM distillation, linking the gains to how foundation-model information is used for correction. These findings show how foundation models can improve an existing lightweight predictor without entering its inference path. Future work will explore adaptive state partitioning and context construction, and the use of graph foundation models to incorporate relational information. We will also investigate how much of the teacher's improvement can be retained by a student that uses only the original features.

AI USAGE DISCLOSURE

Generative AI tools were used for sentence formulation, LaTeX editing, arithmetic checks of tables, drafting mathematical expressions and code, and editing diagram annotations in this manuscript. The LLM used in the research method and the experimental settings are specified in Appendix A.

ETHICAL CONSIDERATIONS

The LLM inputs in our experiments consist of feature descriptions and aggregated errors, without raw values from individual rows. LLM hypotheses about the states are used to select features and examples for prediction, not to make judgments about individuals' actual characteristics or causal explanations. Real-world applications require a separate review of the relevant data-use permissions and privacy requirements.

REPRODUCIBILITY STATEMENT

Dataset identifiers, preprocessing, splitting and cross-fitting procedures, model configurations, and selection procedures are documented in Appendix A. Complete per-dataset results and component-replacement results are provided in Appendices B and C. The accompanying JSON and CSV files preserve the reported experimental scores and aggregate values.

REFERENCES

N. Abhyankar, P. Shojaee, and C. K. Reddy. LLM-FE: Automated feature engineering for tabular data with LLMs as evolutionary optimizers. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=qvI35hkpOO.

A. Arazi, E. Shapira, and R. Reichart. TabSTAR: A tabular foundation model for tabular data with text fields. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

E. Boix-Adserà. Towards a theory of model distillation. *arXiv preprint arXiv:2403.09053*, 2024. doi: 10.48550/arXiv.2403.09053.

G. C. Cawley and N. L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010. URL https://www.jmlr.org/papers/v11/cawley10a.html.

C. Chen, J. Yu, S. Chen, C. Liu, Z. Wan, S. Zhou, Y. Luo, R. Zhang, D. Bitterman, F. Wang, and K. Shu. ClinicalBench: Can LLMs beat traditional ML models in clinical prediction? In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026. doi: 10.1145/3770855.3818831.

T. Chen and C. 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. doi: 10.1145/2939672.2939785.

K. Choi, C. Cundy, S. Srivastava, and S. Ermon. LMPriors: Pre-trained language models as task-specific priors. In *NeurIPS 2022 Workshop on Foundation Models for Decision Making*, 2022.

T. Dao, G. M. Kamath, V. Syrgkanis, and L. Mackey. Knowledge distillation as semiparametric inference. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2104.09732.

Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, UK, 2001. ISBN 978-0-471-87339-6.

T. Dinh, Y. Zeng, R. Zhang, Z. Lin, M. Gira, S. Rajput, J.-y. Sohn, D. Papailiopoulos, and K. Lee. LIFT: Language-interfaced fine-tuning for non-language machine learning tasks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11763–11784, 2022. doi: 10.52202/068431-0855.

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

N. Frosst and G. Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017. doi: 10.48550/arXiv.1711.09784.

J. Gardner, J. C. Perdomo, and L. Schmidt. Large scale transfer learning for tabular data via language modeling. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1435.

Y. Gorishniy, A. Kotelnikov, and A. Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. *arXiv preprint arXiv:2410.24210*, 2024. doi: 10.48550/arXiv.2410.24210.

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

S. Han, J. Yoon, S. Ö. Arik, and T. Pfister. Large language models can automatically engineer features for few-shot tabular learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17454–17479, 2024.

S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag. TabLLM: Few-shot classification of tabular data with large language models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5549–5581. PMLR, 2023. URL https://proceedings.mlr.press/v206/hegselmann23a.html.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. doi: 10.48550/arXiv.1503.02531.

N. Hollmann, S. Müller, and F. Hutter. Large language models for automated data science: Introducing CAAFE for context-aware automated feature engineering. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44753–44775, 2023. doi: 10.52202/075280-1938.

N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, S. B. Hoo, R. T. Schirrmeister, and F. Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 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. *arXiv preprint arXiv:2407.04491*, 2024. URL https://arxiv.org/abs/2407.04491.

D. P. Jeong, Z. C. Lipton, and P. Ravikumar. LLM-Select: Feature selection with large language models. *Transactions on Machine Learning Research*, 2025.

I. Kang, P. Ram, Y. Zhou, H. Samulowitz, and O. Seneviratne. Language model representations for efficient few-shot tabular classification. In *Proceedings of the ACM Web Conference 2026*, 2026. doi: 10.1145/3774904.3792477.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

J. Nam, K. Kim, S. Oh, J. Tack, J. Kim, and J. Shin. Optimized feature generation for tabular data via LLMs with decision tree reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 92352–92380, 2024.

L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. CatBoost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

R. Shi, Y. Wang, M. Du, H. Ye, Y. Chang, and X. Wang. TAROT: Task-adaptive refinement of LLM-prior graphs for few-shot tabular learning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026. doi: 10.1145/3770855.3817944.

D. Slack and S. Singh. TABLET: Learning from instructions for tabular data. *arXiv preprint arXiv:2304.13188*, 2023. doi: 10.48550/arXiv.2304.13188.

M. Spinaci, M. Polewczyk, M. Schambach, and S. Thelin. ConTextTab: A semantics-aware tabular in-context learner. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

R. Wang, Z. Wang, and J. Sun. UniPredict: Large language models are universal tabular classifiers. *arXiv preprint arXiv:2310.03266*, 2023. doi: 10.48550/arXiv.2310.03266.

Y.-S. Wang, K.-Y. Chen, Y.-C. Den, and D. T.-H. Chang. BoostLLM: Boosting-inspired LLM fine-tuning for few-shot tabular classification. *arXiv preprint arXiv:2605.06117*, 2026. doi: 10.48550/arXiv.2605.06117.

Z. Wang, C. Gao, C. Xiao, and J. Sun. MediTab: Scaling medical tabular data predictors via data consolidation, enrichment, and refinement. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 6062–6070, 2024. doi: 10.24963/ijcai.2024/670.

G. Yakushev, A. Shutova, I. Rubachev, N. Bereberdina, R. Sergazinov, and A. Babenko. Talking trees: Reasoning-assisted induction of decision trees for tabular data. *arXiv preprint arXiv:2509.21465*, 2025. doi: 10.48550/arXiv.2509.21465.

J. Yan, B. Zheng, H. Xu, Y. Zhu, D. Z. Chen, J. Sun, J. Wu, and J. Chen. Making pre-trained language models great on tabular prediction. In *The Twelfth International Conference on Learning Representations*, 2024.

J. Yan, J. Chen, C. Hu, B. Zheng, Y. Hu, J. Sun, and J. Wu. Small models are LLM knowledge triggers for medical tabular prediction. In *The Thirteenth International Conference on Learning Representations*, 2025.

H. Ye, J. Li, H. Zhao, D. Guo, and Y. Chang. LLM meeting decision trees on tabular data. *arXiv preprint arXiv:2505.17918*, 2025. doi: 10.48550/arXiv.2505.17918.

E. Zhang, R. Goto, N. Sagan, J. Mutter, N. Phillips, A. Alizadeh, K. Lee, J. Blanchet, M. Pilanci, and R. Tibshirani. LLM-Lasso: A robust framework for domain-informed feature selection and regularization. *arXiv preprint arXiv:2502.10648*, 2025a. doi: 10.48550/arXiv.2502.10648.

E. Zhang, N. Sagan, D. Tse, F. Zhang, M. Pilanci, and J. Blanchet. Learning when to trust LLM priors: A validated framework for semantic prior integration. *arXiv preprint arXiv:2601.21410*, 2026. doi: 10.48550/arXiv.2601.21410.

X. Zhang, G. Ren, H. Yu, et al. LimiX: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025b. doi: 10.48550/arXiv.2509.03505.

M. Zhu, S. Stanivuk, A. Petrović, M. Nikolić, and P. Liò. Incorporating LLM priors into tabular learners. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

A Data and Implementation Settings

Datasets. Table 4 lists the OpenML dataset IDs and actual targets for the 18 evaluated datasets. All classification tasks are binary, and AUROC is computed from the binary ground-truth labels and predicted positive-class probabilities. The Telco identifier `customerID` is excluded from the inputs.

Table 4: Metadata for the OpenML datasets used in the evaluation. p is the number of features actually used as model inputs, excluding targets and identifiers.

Task	OpenML dataset	OpenML ID	n	p	Target
Regression	Diabetes(scikit-learn)	41514	442	10	`class`
	forest_fires	44962	517	12	`area`
	puma8NH	225	8,192	8	`thetadd3`
	rainfall	41539	16,755	3	`Rainfall`
	optical_interconnection_network	42365	640	9	`Channel_Utilization`
	analcatdata_vineyard	500	468	3	`Lugs`
	strikes	549	625	6	`strike_volume`
	analcatdata_apnea1	557	475	3	`Count`
	places	509	329	9	`Economics`
Classification	adult	1590	48,842	14	`class`
	mammography	310	11,183	6	`class`
	telco-customer-churn	42178	7,043	19[†]	`Churn`
	solar-flare	40702	1,066	10	`class`
	pbc	810	418	18	`binaryClass`
	eucalyptus	990	736	19	`binaryClass`
	jEdit_4.2_4.3	1048	369	8	`Bug-count`
	pc1_req	1167	320	8	`DEFECT`
	art_daily	43051	4,032	2	`Class`

[†] `customerID` is excluded from the inputs because it is an identifier.

Algorithm 1 Residual-state teacher and distillation into an X-only student

Input: Fixed $\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}}, \mathcal{D}_{\mathrm{test}}$, maximum stage $H = 5$, $\mathcal{A} = \{0, .25, .5, .75, 1\}$
Output: Final student using only the original inputs X

Partition the row indices of $\mathcal{D}_{\mathrm{tr}}$ into five outer folds $\{\mathcal{I}_l\}_{l=1}^{5}$
for each outer fold $l = 1, \ldots, 5$ **do**
    Let $\mathcal{D}_{-l}$ exclude the rows in $\mathcal{I}_l$; perform the entire teacher-training procedure below using only these data
    Perform preprocessing and F_0 HPO within each fitting context in $\mathcal{D}_{-l}$, and generate inner three-fold predictions
    $\hat{y}_{\mathrm{prev}} \leftarrow$ cross-fitted F_0 predictions on $\mathcal{D}_{-l}$
    for $h = 1, \ldots, H$ **do**
        Construct three states from the cross-fitted errors of $\hat{y}_{\mathrm{prev}}$
        Generate an LLM feature/focus plan using only the aggregate failure audit of $\mathcal{D}_{-l}$
        Generate cross-fitted state representations $\tilde{q}_{r,i}^{(h)}$ using state-specific anchor contexts and LiMiX-2M
        Train correction $\hat{d}_h^{(-l)}$ on the residual target $r_h = y - \hat{y}_{\mathrm{prev}}$
        Construct $T_h^{(-l)}$ by adding the output of $0.25\hat{d}_h^{(-l)}$ to the current predictions
        For classification, clip the corrected probabilities to $[10^{-6}, 1 - 10^{-6}]$
        For $i \in \mathcal{I}_l$, set $t_{h,i}^{\mathrm{OOF}} \leftarrow T_h^{(-l)}(x_i)$
        $\hat{y}_{\mathrm{prev}} \leftarrow$ cross-fitted predictions of the current teacher generated within $\mathcal{D}_{-l}$
    end for
end for
for $h = 1, \ldots, H$ and $\alpha \in \mathcal{A}$ **do**
    $\tilde{y}_h^{(\alpha)} \leftarrow (1 - \alpha)y + \alpha t_h^{\mathrm{OOF}}$
    Train student S_α using only the original X
    Score by minimizing validation RMSE (regression) or maximizing validation AUROC (classification)
end for
Select $(\hat{h}, \hat{\alpha})$ by the student's validation metric and evaluate on the independent test split

Multiclass extension. An extension to multiclass classification can blend one-hot labels with the teacher's probability vectors using the same weights and train with cross-entropy. All classification tasks in our experiments are binary.

Execution order. Algorithm 1 selects the final candidate based on the validation performance of students trained for each combination of stage and mixing weight, rather than fixing the teacher stage in advance. In the algorithm, $\tilde{y}_h^{(\alpha)}$ is the blended target vector for distillation and is distinct from the residual-state probability representation q_r. The parameter α is the teacher weight.

Table 5 summarizes data splitting and base-model training, while Table 6 summarizes the proposed method's fixed settings and student selection. The base model, correction model, and student reuse the XGBoost hyperparameters and boosting rounds selected within the same fitting context, each fitted to its own training target. Model-specific HPO and additional selection budgets are given in Table 7, and detailed search spaces are provided in Tables 8 and 9.

Table 5: Data splitting and base-model training settings for the final experiments on 18 datasets.

Item	Setting
Repetitions and splits	Both regression and classification use seeds $\{2032, 2033, 2034\}$. For each seed, the train/-validation/test ratio is 60/20/20. Regression uses duplicate-feature group-based splitting to prevent identical input X rows from appearing in different splits. Classification uses class-stratified splitting.
OOF	Outer five-fold and inner three-fold cross-fitting are used, with GroupKFold for regression and StratifiedKFold for classification. Each outer fold's teacher construction and all iterative stages are performed within the corresponding outer-training data. Each training row has exactly one outer-OOF teacher prediction per stage.
Preprocessing	Preprocessing is fitted only on the training rows in each fitting context. Missing numerical values are imputed with the training median. Missing categorical values are represented as `__MISSING__`, and one-hot encoding uses the training vocabulary. Unseen categories receive an all-zero encoding. No additional numerical scaling or target transformation is applied.
Base F_0 HPO	XGBoost is used. Each fitting context searches over 32 candidates: one default and 31 deterministic Latin-hypercube candidates. HPO uses a separate 80:20 split within the corresponding training data, with a maximum of 3,000 boosting rounds and early stopping after 100 rounds. Search ranges are given in Table 8.
F_0 selection and reuse	The candidate minimizing internal HPO validation RMSE is selected for regression, and the candidate maximizing internal HPO validation AUROC is selected for classification. The model is refitted on the entire fitting context using the selected hyperparameters and boosting rounds. The correction teacher and student also reuse these F_0 settings and are refitted to their respective targets.
Uncertainty	$\pm$ denotes the sample standard deviation at the aggregation level specified in each table or figure. Per-dataset performance uses the standard deviation of scores across three seeds, while rank distributions use the standard deviation of ranks across datasets.
Computing environment	The environment uses Linux 5.15, Python 3.13.14, NumPy 2.3.5, pandas 2.3.3, scikit-learn 1.7.2, SciPy 1.16.3, and PyTorch 2.12.0. Regression uses bundled XGBoost 3.2.0, and classification uses bundled XGBoost 3.4.1. The GPU is an NVIDIA RTX PRO 6000 Blackwell (97,887 MiB VRAM), and tree runtimes are limited to 4 CPU threads.

Fixed settings and final selection for the proposed method. The correction coefficient $\gamma = 0.25$, number of states $K = 3$, classification margin $\delta = 0.2$, and total ICL context budget of at most 256 rows are fixed. No separate HPO is performed for the LLM or TFM. These values are fixed globally across datasets rather than tuned separately for individual datasets. After HPO for F_0, the final student is selected by validation performance over combinations of stage $h \in \{1, 2, 3, 4, 5\}$ and teacher mixing weight $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$.

Table 6: Final fixed settings and student-selection procedure for the proposed method. Ours in the main text denotes the Qwen-based final student.

Item	Setting
Regression states	Deterministic one-dimensional k-means is applied to the current stage's signed residuals $r = y - \hat{y}_{\mathrm{prev}}$. The number of states is fixed at $K = 3$.
Classification states	The true-class margin $m = p(y) - p(1-y)$ is partitioned into three states by fixed-margin binning with threshold $\delta = 0.2$: $S_1 : m < -0.2$, $S_2 : \lvert m \rvert \leq 0.2$, and $S_3 : m > 0.2$.
LLM	The main results use `Qwen/Qwen3.5-122B-A10B-GPTQ-Int4` through local vLLM, with temperature $= 0.1$, top-$p = 0.9$, max tokens $= 1024$, request attempts $= 2$, workers $= 16$, and `enable_thinking=False`. The model and inference settings are fixed, with no separate HPO.
LLM inputs	No raw row values are provided. Inputs consist only of feature names, data types, numerical quantiles, missing-value counts, the most frequent categories, baseline error/residual summaries, and high-error subgroups. The LLM returns up to eight features and a verifiable focus condition in JSON format.
ICL context	State-specific anchors are used, with a total context budget of at most 256 rows. At least one example is included from each nonempty state, and the remaining examples are sampled according to the natural state frequencies. The query's original y is not provided through the LiMiX input channel.
TFM	A fixed LiMiX-2M checkpoint is used without separate HPO. Settings are fixed to GPU mixed precision, a query batch size of 512, and no query-row deduplication.
Teacher	The method uses at most $H = 5$ stages and a fixed correction coefficient $\gamma = 0.25$. The correction target is $y - \hat{y}_{\mathrm{prev}}$, and the teacher is constructed as $\hat{y}_{\mathrm{prev}} + \gamma \hat{d}_h$. Classification probabilities are clipped to $[10^{-6}, 1 - 10^{-6}]$ after each stage. Correction-model inputs are $[X, q_r]$ for regression and $[X, \hat{y}_{\mathrm{prev}}, q_r]$ for classification. The X-only control removes only q_r, keeping all other settings unchanged.
Student	The final student is an XGBoost model using only the original X. For $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$, it is trained on targets $(1-\alpha)y + \alpha t_h^{\mathrm{OOF}}$. Stage $h \in \{1, 2, 3, 4, 5\}$ and α are jointly selected by the student's validation metric, not the teacher's performance: minimum RMSE for regression and maximum AUROC for classification. Ties favor earlier stages and smaller α.

No-LLM control. In the no-LLM condition in Table 2, the list of feature names is randomly shuffled according to the seed at each stage and outer fold, and up to eight features are selected. The feature count is not matched to the LLM plan; therefore, all features are used when there are eight or fewer. We set `focus_condition=None` while retaining the three-state partition. Following the original procedure, at least one anchor is selected per state, and the remaining anchors are chosen randomly, for a total of at most 256. Thus, the no-LLM control removes both LLM-guided feature selection and focus-based anchor prioritization while retaining the three-state partition and the same context budget.

Direct TFM $\rightarrow$ student control. Using the same five outer folds, we directly predict the original target Y for the held-out portion of each fold, providing only that fold's training portion as context. Fold-specific predictions are combined into a single-column OOF teacher target $\hat{Y}_{\mathrm{TFM,OOF}}$, in which every training row is predicted exactly once. The context budget is at most 256 rows per fold. For regression, context rows are sampled randomly, the actual Y values are standardized within each context, and predictions from the LiMiX regression head are transformed back to the original target scale. For classification, context sampling follows the proportions of the two original classes while including at least one row from each class, and LiMiX's positive-class probabilities are used. The student is an XGBoost model that receives only the original X from the full training set, with training targets $\tilde{Y}^{(\alpha)} = (1-\alpha)Y + \alpha \hat{Y}_{\mathrm{TFM,OOF}}$, $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. Among the five α

candidates, we select the one minimizing validation RMSE for regression or maximizing validation AUROC for classification. XGBoost parameters and boosting rounds are reused from the existing standalone F_0 HPO selection. Because Direct TFM has no iterative stages, only five candidates are compared: one stage $\times$ five α values.

Model-specific HPO and selection budgets. The baselines are XGB, CatBoost, LGBM, TabM, RealMLP, TabPFN3, LiMiX-2M, OCTree, and DeLTa. XGB, CatBoost, LGBM, TabM, and RealMLP each use 32 candidates. Pretrained models and LLM-based baselines use the fixed settings or separate selection procedures listed in Table 7.

Table 7: Model-specific HPO and selection budgets actually used for the final results. A 32-trial budget refers to the number of candidates in the corresponding HPO and does not imply equal total numbers of training runs or equal computational budgets across methods.

Model	HPO / selection budget	Applied settings
Ours	32 trials per F_0; then stage and α selection	Selected XGB hyperparameters and boosting rounds are reused for the correction model and student. Selection is over stages $\{1, 2, 3, 4, 5\}$ and α values $\{0, 0.25, 0.5, 0.75, 1\}$.
XGB	32 trials	One default candidate + 31 Latin-hypercube candidates. Maximum 3,000 rounds; early stopping after 100 rounds.
CatBoost	32 trials	One default candidate + 31 Latin-hypercube candidates. Maximum 3,000 rounds; early stopping after 300 rounds for regression / 100 for classification.
LGBM	32 trials	One default candidate + 31 Latin-hypercube candidates. Maximum 3,000 rounds; early stopping after 300 rounds for regression / 100 for classification.
TabM	32 trials	One default candidate + 31 TabArena candidates from `pytabkit 1.7.3`. Early-stopping patience is 16.
RealMLP	32 trials	One TD default candidate + 31 TabArena candidates. 256 / 512 epochs per candidate; early stopping is applied to the extended variant.
TabPFN3	No separate HPO	Default and automatic ensemble settings of the V3 pretrained model are used.
LiMiX-2M	No separate HPO	A fixed checkpoint and inference settings are used. Retrieval is enabled for regression and disabled for classification.
OCTree	Reuses tuned XGB settings; 50 LLM optimization iterations	One generated feature; feedback CART depth $\{1, 2, 3, 4\}$. Selection uses MAE for regression / AUROC for classification.
DeLTa	No separate RF HPO; separate classification η setting	Regression: RF 5, depth 10, 200 leaves, 10 rules, TabPFN leaf. Classification: RF 20 (15 for Adult), depth 20, 50 leaves, 10 rules, CART leaf.

Differences in validation data and tuning procedures. XGB's 32-trial HPO uses a separate 80:20 split within the training data. In contrast, CatBoost, LGBM, TabM, and RealMLP use the external validation split. Ours performs 32-trial HPO for F_0 in each fitting context and then selects the stage and α combination on the external validation set according to final student performance. Thus, what is matched in this comparison is the number of HPO candidates for the five baselines above and for each F_0, not the use of validation data or the entire tuning procedure, including additional selection. In particular, OCTree's LLM optimization and DeLTa's η selection are distinct from 32-trial HPO.

Classification η selection for DeLTa. For Adult, $\eta = 0.12$ is fixed. For the remaining eight classification datasets, we select the value that maximizes validation AUROC from the following candidates.

$$\eta \in \{0,\ 0.01,\ 0.02,\ 0.024,\ 0.04,\ 0.044,\ 0.06, 0.08,\ 0.10,\ 0.12,\ 0.15,\ 0.18,\ 0.20,\ 0.25\}.$$

Search spaces for tree models. Table 8 gives the search ranges used for the 31 Latin-hypercube candidates of XGB, CatBoost, and LGBM. One default candidate for each model is included separately from these sampling ranges. In the table, log denotes logarithmic-scale search.

Table 8: Final HPO search ranges for the tree models. These ranges apply to the 31 search candidates, excluding the default candidate.

Parameter	XGB	CatBoost	LGBM
Learning rate	0.01–0.2, log	0.01–0.2, log	0.01–0.2, log
Depth	$\{3, 4, 5, 6, 8\}$	Integers 4–10	$\{-1, 4, 6, 8, 10, 12\}$
Num leaves	—	—	Integers 8–255
Minimum leaf constraints	child weight 0.5–20, log	min data 1–63	min child samples 5–99
Row sampling	subsample 0.6–1.0	bagging temperature 0–5	subsample 0.6–1.0
Column sampling	0.6–1.0	rsm 0.6–1.0	0.6–1.0
L1	0 or 10^{-4}–10, log	—	10^{-5}–10, log
L2	10^{-3}–100, log	10^{-4}–100, log	10^{-4}–100, log
Other	`tree_method=hist`	random strength 0–5	subsample frequency: 0 for regression / 1 for classification

For regression, LGBM uses `subsample_freq=0`, so row subsampling is disabled even though the table includes a subsample search range. Classification uses `subsample_freq=1`.

Search spaces for TabM and RealMLP. Table 9 gives the search spaces for the 31 candidates excluding the default. TabM uses one default candidate and 31 TabArena candidates from `pytabkit 1.7.3`, while RealMLP uses one TD default candidate and 31 TabArena candidates. Both models use `n_cv=1` and `n_refit=0`, with an explicit external validation set. Candidates are selected by minimizing RMSE for regression and maximizing AUROC for classification.

Table 9: Final HPO search ranges for TabM and RealMLP. These apply to the 31 candidates other than the default; log denotes logarithmic-scale search.

Item	TabM	RealMLP
Architecture	`tabm-mini`, 32 members	hidden layers $\{2, 3, 4\}$
Width	block dimension 128–1024, in increments of 16	hidden width $\{256, 384, 512\}$
Number of blocks	$\{2, 3, 4, 5\}$	—
Learning rate	10^{-4}–3×10^{-3}, log	0.02–0.3, log
Weight decay	0 or 10^{-4}–0.1, log	0.001–0.05, log
Dropout	0 or 0–0.5	0–0.5
Numerical embeddings	PWL, dimension $\{8, 12, \ldots, 32\}$, bins 2–128	PLR sigma 0.01–50, log
Additional learning-rate factors	—	PLR 0.05–0.3, scale 2–10, first layer 0.3–1.5; all log
Adam β_2	Default setting	$1 - \mathrm{LogUniform}(0.005, 0.05)$
Label smoothing	—	on/off for classification, ε 0.005–0.1, log
Stopping criterion	early-stopping patience 16	Default variant: 256 epochs; extended variant: 256 / 512 epochs with early stopping

For the extended RealMLP variant, each PLR hidden size is selected from $\{8, 16, 32, 64\}$. Early stopping for the extended variant uses a multiplicative patience of 3 and an additive patience of 40.

A.1 Example Qwen Prompts and Recorded Output

The following system prompt, user input, and response form one recorded interaction from the Qwen execution log for `art_daily_nojump4` (OpenML 43051), seed 2032, stage $h = 3$, `fold_3`. The user input contains the schema and aggregated errors from 1,935 rows in the corresponding training fold, and the source of the previous-stage predictions is recorded as `inner_cross_fitted_previous_stage`.

System prompt.

```
You design one virtual failure-state view for a binary tabular classifier.
Return one strict JSON object and no markdown. Use only feature names and
  candidate
conditions present in the supplied aggregate training-fold audit. Never infer
  or request
individual rows. The operational latent target will be a three-level
  fixed-margin bin of the
cross-fitted true-class probability margin, p(true class) - p(other class): low
(confidently wrong), middle (near the decision boundary), high (correct with
  confidence). Select 3-8
source features that should help a tabular foundation model infer that state.
  Select at
most one focus condition from the supplied subgroup list; it only prioritizes
  anchors.
Do not include the task target as a feature. Required schema:
{"name":"short_identifier","hypothesis":"one sentence","features":["..."],
 "focus_condition":null OR
  {"feature":"...","operator":"<=|>|==|!=","value":...},
 "classes":3}
```

User prompt.


```
{
  "dataset": "art_daily_nojump4",
  "training_fold_failure_audit": {
    "rows": 1935,
    "positive_rate": 0.03255813953488372,
    "baseline": {
      "mean_probability": 0.0345910652387547,
      "error_rate": 0.03255813953488372,
      "mean_absolute_residual": 0.06459197255958234
    },
    "schema": [
      {
        "feature": "timestamp",
        "type": "categorical",
        "top_categories": ["2014-04-01 00:00:00", "2014-04-01 00:05:00",
  "2014-04-01 00:20:00", "2014-04-01 00:25:00", "2014-04-01 00:30:00",
  "2014-04-01 00:35:00", "2014-04-01 00:50:00", "2014-04-01 01:05:00",
  "2014-04-01 02:05:00", "2014-04-01 02:15:00", "2014-04-01 02:20:00",
  "2014-04-01 02:25:00"],
        "distinct": 1935
      },
      {
        "feature": "value",
        "type": "numeric",
        "quantiles": [18.62526363108, 19.6702488807, 21.3876535631,
  74.3139687787, 81.9019500224],
        "missing": 0
      }
    ],
    "highest_residual_subgroups": [
      {
        "feature": "value",
        "operator": ">",
        "value": 74.3139687787,
        "rows": 484,
```

```
        "mean_residual": -0.025300226100453,
        "mean_absolute_residual": 0.025300226100453,
        "error_rate": 0.0
      },
      {
        "feature": "value",
        "operator": "<=",
        "value": 19.6702488807,
        "rows": 484,
        "mean_residual": 0.017849393974786888,
        "mean_absolute_residual": 0.10844879423653345,
        "error_rate": 0.06611570247933884
      }
    ],
    "row_values_included": false,
    "residual_state_classes": 3,
    "correction_stage": 3,
    "prediction_source": "inner_cross_fitted_previous_stage",
    "state_target": "true_class_margin=p(true_class)-p(other_class), delta=0.2"
  }
}
```


Recorded output.


```
{
  "name": "art_daily_nojump4_failure_view",
  "hypothesis": "The model exhibits high uncertainty or confident errors when
  the observed value is low, particularly below the 25th percentile, while high
  values are consistently predicted correctly.",
  "features": ["value", "timestamp"],
  "focus_condition": {
    "feature": "value",
    "operator": "<=",
    "value": 19.6702488807
  },
  "classes": 3
}
```


In this response, Qwen selects `value` and `timestamp` and returns a hypothesis about the low-`value` region, together with the condition $\texttt{value} \leq 19.6702488807$. As specified in the system prompt, the focus condition only prioritizes anchor selection; the response itself is not a sample-specific state label, a residual-state probability, or a residual correction.

B Complete Comparison Results on 18 Datasets

Table 10 summarizes mean performance and average ranks by task, while Tables 11 and 12 report per-dataset means and standard deviations. Ours denotes the final student, which uses only the original observed features. Lower ranks are better, and bold values indicate the best displayed mean (lowest for NRMSE and highest for AUROC). Each cell in the detailed per-dataset tables reports both the mean and standard deviation.

Table 10: Summary of the main results on 18 datasets. Results are aggregated separately for nine regression and nine classification datasets. Average ranks for each task are recomputed from the per-dataset mean performance. The overall rank is the equally weighted average of ranks across all 18 datasets, rounded only at the end, and is equivalent to the equally weighted average of the two unrounded task-specific average ranks.

	Regression		Classification		Overall
Model	Rank ↓	NRMSE ↓	Rank ↓	AUROC ↑	Rank ↓
Ours (student)	**4.11**	0.8232	**3.00**	0.7857	**3.56**
TabM	4.22	0.8150	3.78	**0.7915**	4.00
TabPFN3	4.56	0.8514	4.00	0.7849	4.28
RealMLP	4.33	**0.8114**	7.67	0.7589	6.00
LiMiX-2M	6.00	0.8486	5.67	0.7807	5.83
OCTree	5.78	0.8434	5.56	0.7746	5.67
LGBM	6.67	0.9098	4.44	0.7785	5.56
CatBoost	6.22	0.8305	5.78	0.7743	6.00
XGB	6.78	0.8399	6.89	0.7703	6.83
DeLTa	6.33	0.8544	8.22	0.7441	7.28

Table 11: NRMSE results on the nine regression datasets.

Method	Diabetes	Forest	Puma8NH	Rainfall	Optical	Vineyard	Strikes	Apnea1	Places
Ours	0.8155 ±0.0383	**2.4666** ±2.1049	0.5769 ±0.0194	0.4823 ±0.0061	0.3459 ±0.2408	0.3916 ±0.0706	1.1727 ±0.6428	0.2893 ±0.0734	0.8679 ±0.0432
TabM	0.7663 ±0.0517	2.4791 ±2.1119	0.5654 ±0.0192	**0.4401** ±0.0143	**0.2394** ±0.2956	0.3582 ±0.0514	1.1912 ±0.6200	0.3945 ±0.1829	0.9010 ±0.0363
RealMLP	0.7538 ±0.0740	2.5072 ±2.1215	**0.5600** ±0.0151	0.4505 ±0.0176	0.2700 ±0.3123	0.3715 ±0.0443	1.1982 ±0.6358	0.3098 ±0.0703	0.8810 ±0.0598
TabPFN3	**0.7442** ±0.0532	2.4869 ±2.1183	0.5607 ±0.0164	0.5129 ±0.0192	0.5917 ±0.7952	**0.3527** ±0.0516	**1.1019** ±0.7053	0.4704 ±0.1036	**0.8414** ±0.0345
OCTree	0.8605 ±0.0601	2.4789 ±2.1240	0.5724 ±0.0164	0.4923 ±0.0184	0.4051 ±0.2709	0.4038 ±0.0519	1.1599 ±0.6367	0.3447 ±0.1067	0.8728 ±0.0611
LiMiX-2M	0.8032 ±0.0264	2.4813 ±2.1191	0.5957 ±0.0167	0.5129 ±0.0135	0.5878 ±0.9288	0.3923 ±0.0607	1.1094 ±0.6948	**0.2664** ±0.0591	0.8886 ±0.0291
CatBoost	0.8030 ±0.0733	2.4826 ±2.1058	0.5682 ±0.0171	0.5328 ±0.0097	0.2961 ±0.3135	0.4112 ±0.0782	1.1815 ±0.6477	0.3013 ±0.0217	0.8981 ±0.0758
DeLTa	0.8126 ±0.0466	2.4959 ±2.1356	0.5654 ±0.0118	0.5089 ±0.0107	0.5497 ±0.2449	0.3987 ±0.0354	1.1341 ±0.6747	0.3037 ±0.0925	0.9204 ±0.0621
LGBM	0.7780 ±0.0730	2.4778 ±2.1105	0.5759 ±0.0187	0.4829 ±0.0095	0.4654 ±0.2566	0.4231 ±0.0643	1.1764 ±0.6269	0.8661 ±0.1131	0.9423 ±0.0667
XGB	0.8549 ±0.0488	2.4792 ±2.1120	0.5783 ±0.0188	0.4832 ±0.0055	0.3477 ±0.2382	0.4039 ±0.0619	1.2142 ±0.5934	0.3185 ±0.0878	0.8794 ±0.0412

Table 12: AUROC results on the nine classification datasets.

Method	Adult	Mammo-graphy	Telco-Churn	Solar-Flare	PBC	Eucalyptus	jEdit	pc1_req	art_daily
Ours	**0.9312** ±0.0023	0.9528 ±0.0098	0.8484 ±0.0188	0.7426 ±0.0417	0.7628 ±0.0610	0.8491 ±0.0201	0.6918 ±0.0780	0.5767 ±0.0872	0.7163 ±0.0623
TabM	0.9287 ±0.0026	0.9448 ±0.0329	0.8485 ±0.0157	0.7320 ±0.0519	0.7714 ±0.0623	0.8450 ±0.0300	0.6849 ±0.0807	**0.6282** ±0.0364	**0.7399** ±0.0754
TabPFN3	0.9185 ±0.0025	0.9567 ±0.0089	0.8228 ±0.0145	**0.7453** ±0.0305	**0.7729** ±0.0595	**0.8625** ±0.0166	**0.7036** ±0.0510	0.5663 ±0.0528	0.7153 ±0.0723
LGBM	0.9305 ±0.0016	0.9480 ±0.0148	**0.8495** ±0.0164	0.7377 ±0.0320	0.7702 ±0.0624	0.8356 ±0.0208	0.6731 ±0.0897	0.5368 ±0.0562	0.7252 ±0.0786
OCTree	0.9299 ±0.0020	0.9472 ±0.0126	0.8437 ±0.0165	0.7279 ±0.0458	0.7451 ±0.0617	0.8465 ±0.0164	0.6413 ±0.0701	0.6110 ±0.0721	0.6783 ±0.0876
CatBoost	0.9309 ±0.0019	0.9427 ±0.0145	0.8492 ±0.0144	0.7128 ±0.0385	0.7206 ±0.0283	0.8456 ±0.0178	0.6881 ±0.0644	0.5702 ±0.0183	0.7086 ±0.0697
LiMiX-2M	0.9183 ±0.0012	**0.9615** ±0.0106	0.8468 ±0.0155	0.7242 ±0.0461	0.7681 ±0.0559	0.8448 ±0.0087	0.6876 ±0.0767	0.5601 ±0.1020	0.7146 ±0.0706
XGB	0.9308 ±0.0021	0.9437 ±0.0164	0.8471 ±0.0177	0.7265 ±0.0243	0.7371 ±0.0710	0.8230 ±0.0280	0.6364 ±0.0848	0.5745 ±0.0858	0.7135 ±0.0599
RealMLP	0.9283 ±0.0021	0.9467 ±0.0025	0.8462 ±0.0170	0.6998 ±0.0842	0.7374 ±0.0695	0.8347 ±0.0169	0.6292 ±0.0403	0.5798 ±0.0374	0.6276 ±0.1210
DeLTa	0.9115 ±0.0021	0.9383 ±0.0203	0.8443 ±0.0141	0.7026 ±0.0443	0.7418 ±0.0431	0.8322 ±0.0179	0.6375 ±0.0192	0.5807 ±0.0832	0.5085 ±0.0153

C Detailed Comparisons of Representation Estimation Modules

LLM replacement. LLM replacement experiments are conducted on the nine regression datasets. Table 13 maps short names to model identifiers, and Table 14 reports the final student's NRMSE after each replacement.

Table 13: Model identifiers for the LLM replacement experiments.

Short name	Supplied model identifier
Q3.5-122B	`Qwen3.5-122B-A10B-GPTQ-Int4`
Q3.5-35B	`Qwen3.5-35B-A3B`
Q3.5-9B	`Qwen3.5-9B`
Q2.5-72B	`Qwen2.5-72B-Instruct-GPTQ-Int4`

Table 14: LLM replacement experiments: complete regression NRMSE results. Table 13 maps the short names to model identifiers. Each column reports the final student's results, not the LLM's own direct regression performance.

Dataset	Q3.5-122B	Q3.5-35B	Q3.5-9B	Q2.5-72B
Diabetes	$\mathbf{0.8155}_{\pm 0.0383}$	$0.8229_{\pm 0.0317}$	$0.8354_{\pm 0.0313}$	$0.8384_{\pm 0.0311}$
Forest	$\mathbf{2.4666}_{\pm 2.1049}$	$2.4668_{\pm 2.1012}$	$2.4682_{\pm 2.1040}$	$2.4703_{\pm 2.1014}$
Puma8NH	$0.5769_{\pm 0.0194}$	$\mathbf{0.5763}_{\pm 0.0189}$	$0.5767_{\pm 0.0171}$	$0.5766_{\pm 0.0171}$
Rainfall	$0.4823_{\pm 0.0061}$	$0.4822_{\pm 0.0059}$	$0.4822_{\pm 0.0062}$	$\mathbf{0.4820}_{\pm 0.0065}$
Optical	$\mathbf{0.3459}_{\pm 0.2408}$	$0.3504_{\pm 0.2343}$	$0.3524_{\pm 0.2315}$	$0.3505_{\pm 0.2341}$
Vineyard	$0.3916_{\pm 0.0706}$	$\mathbf{0.3914}_{\pm 0.0712}$	$0.3935_{\pm 0.0712}$	$\mathbf{0.3914}_{\pm 0.0712}$
Strikes	$1.1727_{\pm 0.6428}$	$\mathbf{1.1703}_{\pm 0.6354}$	$1.1891_{\pm 0.6576}$	$1.1717_{\pm 0.6392}$
Apnea1	$\mathbf{0.2893}_{\pm 0.0734}$	$0.3045_{\pm 0.0867}$	$0.3055_{\pm 0.0874}$	$0.3045_{\pm 0.0867}$
Places	$\mathbf{0.8679}_{\pm 0.0432}$	$0.8696_{\pm 0.0413}$	$0.8696_{\pm 0.0413}$	$0.8696_{\pm 0.0413}$

State predictor replacement. Tables 15 and 16 report the final student's performance after replacing LiMiX-2M with XGBoost, random forest, or logistic regression. State prediction is an auxiliary classification task even for regression, because the state labels are categorical. The final student remains XGBoost in every condition.

Table 15: Complete NRMSE results for the state predictor replacement experiments. RF and LR denote random forest and logistic regression, respectively. Because state prediction is an auxiliary classification task even for regression, LR can be used as a control.

Dataset	LiMiX-2M	XGB-state	RF-state	LR-state
Diabetes	$\mathbf{0.8155}_{\pm 0.0383}$	$0.8362_{\pm 0.0306}$	$0.8279_{\pm 0.0184}$	$0.8379_{\pm 0.0329}$
Forest	$\mathbf{2.4666}_{\pm 2.1049}$	$2.4841_{\pm 2.0948}$	$2.4781_{\pm 2.1149}$	$2.4822_{\pm 2.1101}$
Puma8NH	$0.5769_{\pm 0.0194}$	$\mathbf{0.5767}_{\pm 0.0174}$	$0.5772_{\pm 0.0170}$	$0.5772_{\pm 0.0177}$
Rainfall	$0.4823_{\pm 0.0061}$	$0.4828_{\pm 0.0061}$	$\mathbf{0.4822}_{\pm 0.0067}$	$0.4832_{\pm 0.0051}$
Optical	$\mathbf{0.3459}_{\pm 0.2408}$	$0.3477_{\pm 0.2382}$	$0.3600_{\pm 0.2208}$	$0.3464_{\pm 0.2400}$
Vineyard	$0.3916_{\pm 0.0706}$	$\mathbf{0.3888}_{\pm 0.0707}$	$0.3947_{\pm 0.0651}$	$0.3930_{\pm 0.0661}$
Strikes	$\mathbf{1.1727}_{\pm 0.6428}$	$1.1856_{\pm 0.6489}$	$1.1874_{\pm 0.6733}$	$1.1808_{\pm 0.6356}$
Apnea1	$0.2893_{\pm 0.0734}$	$\mathbf{0.2808}_{\pm 0.0764}$	$0.3079_{\pm 0.0845}$	$0.2994_{\pm 0.1067}$
Places	$\mathbf{0.8679}_{\pm 0.0432}$	$0.8784_{\pm 0.0344}$	$0.8743_{\pm 0.0371}$	$0.8741_{\pm 0.0400}$

Table 16: Complete AUROC results for the state predictor replacement experiments. RF and LR denote random forest and logistic regression, respectively.

Dataset	LiMiX-2M	XGB-state	RF-state	LR-state
Adult	$0.9312_{\pm 0.0023}$	$\mathbf{0.9313}_{\pm 0.0019}$	$0.9310_{\pm 0.0021}$	$0.9311_{\pm 0.0022}$
mammography	$\mathbf{0.9528}_{\pm 0.0098}$	$0.9475_{\pm 0.0137}$	$0.9472_{\pm 0.0145}$	$0.9487_{\pm 0.0119}$
telco-churn	$\mathbf{0.8484}_{\pm 0.0188}$	$0.8478_{\pm 0.0183}$	$0.8478_{\pm 0.0183}$	$0.8478_{\pm 0.0183}$
solar-flare	$0.7426_{\pm 0.0417}$	$0.7402_{\pm 0.0392}$	$\mathbf{0.7451}_{\pm 0.0407}$	$0.7448_{\pm 0.0402}$
pbc	$\mathbf{0.7628}_{\pm 0.0610}$	$0.7462_{\pm 0.0704}$	$0.7580_{\pm 0.0679}$	$0.7602_{\pm 0.0650}$
eucalyptus	$\mathbf{0.8491}_{\pm 0.0201}$	$0.8379_{\pm 0.0281}$	$0.8428_{\pm 0.0195}$	$0.8441_{\pm 0.0184}$
jEdit	$\mathbf{0.6918}_{\pm 0.0780}$	$0.6909_{\pm 0.0682}$	$0.6876_{\pm 0.0636}$	$0.6734_{\pm 0.0894}$
pc1_req	$0.5767_{\pm 0.0872}$	$0.5785_{\pm 0.0931}$	$\mathbf{0.5811}_{\pm 0.0907}$	$0.5792_{\pm 0.0936}$
art_daily	$\mathbf{0.7163}_{\pm 0.0623}$	$0.7124_{\pm 0.0590}$	$0.7147_{\pm 0.0609}$	$0.7114_{\pm 0.0583}$

D Computational Cost Comparison

Table 17 compares training/adaptation time and the final predictor's inference latency across three representative datasets and three seeds. By distinguishing representation estimation during training from final prediction, we evaluate the deployment structure in which RECAST uses large models during training but relies on the student alone at inference time. The student's per-sample inference latency is comparable to that of the base XGBoost model in both tasks.

Table 17: Computational cost comparison. Training/adaptation time includes the complete procedure for each method, while inference time is the final predictor's per-sample latency. Values are means±standard deviations for training and means for inference, measured over three representative datasets and three seeds.

Model	Regression		Classification	
	Training/adaptation (s) ↓	Inference (ms/sample) ↓	Training/adaptation (s) ↓	Inference (ms/sample) ↓
Ours (student)	154.62 ± 130.22	$\mathbf{0.0144}$	159.26 ± 160.26	0.0060
TabM	459.06 ± 686.29	0.0533	703.91 ± 818.18	0.0402
RealMLP	403.12 ± 601.45	0.0985	1223.86 ± 1584.05	0.0609
OCTree	505.06 ± 150.44	0.0666	1358.17 ± 1665.62	0.0491
DeLTa	158.03 ± 102.02	0.1650	25.33 ± 2.28	0.0653
TabPFN3	$\mathbf{4.14 \pm 0.56}$	1.2755	242.53 ± 408.66	1.0624
LiMiX-2M	21.81 ± 14.18	27.7768	500.83 ± 863.57	27.7777
LGBM	10.84 ± 12.35	0.0226	17.56 ± 21.80	0.0119
CatBoost	61.15 ± 76.14	0.0207	73.58 ± 101.25	$\mathbf{0.0055}$
XGB	4.69 ± 5.91	0.0146	$\mathbf{6.81 \pm 10.06}$	0.0065

E Evaluation Metrics and Aggregation

NRMSE and AUROC. For regression, to compare performance across datasets with different scales, we normalize test RMSE by the target standard deviation computed on the training split for the corresponding seed. For dataset d, let the index sets of samples in the training and test splits be $\mathcal{I}_d^{\text{train}}$ and $\mathcal{I}_d^{\text{test}}$, respectively. We omit the seed subscript for brevity. NRMSE for dataset d is defined as

$$\text{NRMSE}_d(f) = \frac{\sqrt{|\mathcal{I}_d^{\text{test}}|^{-1} \sum_{i \in \mathcal{I}_d^{\text{test}}} \left(y_i - f(x_i)\right)^2}}{\sqrt{|\mathcal{I}_d^{\text{train}}|^{-1} \sum_{i \in \mathcal{I}_d^{\text{train}}} \left(y_i - \bar{y}_{\text{train},d}\right)^2}}.$$

Here, $\bar{y}_{\text{train},d}$ is the mean target value in the corresponding training split. Within each dataset–seed combination, the same normalization constant in the denominator is used for every model and stage.

The normalization therefore does not change with the residual target or intermediate teacher. No additional target transformation is applied for regression.

For binary classification, AUROC is computed from the binary ground-truth labels and the positive-class probabilities output by each model. Blended targets are used only for student training; performance is always evaluated against the original ground-truth labels.

Repetitions and aggregation. Per-dataset scores and standard deviations are computed from test scores obtained with three split seeds. Mean performance for each task is the equally weighted average of the mean scores on its nine datasets, and the between-dataset standard deviation is the sample standard deviation of those nine means. For each dataset, models are ranked by their mean test performance across seeds; these ranks are then averaged with equal weight across datasets. The overall average rank across regression and classification also gives equal weight to each dataset; intermediate averages are not rounded, and only the final aggregate is rounded. Mean performance and rank in the tables summarize different aspects of performance, and the aggregate performance claims in the main text are based on average rank.

Changes relative to the base model. The relative reduction in NRMSE for regression and the relative increase in AUROC for classification are defined, respectively, as follows.

$$\Delta^{\mathrm{rel}}_{d,\mathrm{reg}} = 100\frac{\mathrm{NRMSE}_d(F_0) - \mathrm{NRMSE}_d(S)}{\mathrm{NRMSE}_d(F_0)},\ \Delta^{\mathrm{rel}}_{d,\mathrm{cls}} = 100\frac{\mathrm{AUROC}_d(S) - \mathrm{AUROC}_d(F_0)}{\mathrm{AUROC}_d(F_0)}$$

Table 1 in the main text uses relative reductions for regression and relative increases for classification. The reduction in task-level mean NRMSE is computed as a ratio of mean values and is distinct from the average of per-dataset reduction rates.

Performance gap in the efficiency plots. We use the performance gap to summarize the different regression and classification metrics in terms of relative error gaps. Let $e_{m,d}$ denote the error of model m on dataset d. We define $\mathrm{Gap}(m) = \frac{100}{9}\sum_{d=1}^{9}\frac{e_{m,d}-e_d^{\mathrm{best}}}{e_{m,d}}$. For regression, we use $e_{m,d} = \mathrm{NRMSE}_{m,d}$, and for classification, we use $e_{m,d} = 1 - \mathrm{AUROC}_{m,d}$ as an AUROC-based error measure. Here, e_d^{best} is the smallest error among the ten models compared on that dataset. For each dataset, errors and gaps are first computed from the mean performance across three seeds, and the results are then averaged with equal weights over the nine datasets.

The performance gap therefore does not represent a relative difference in NRMSE or AUROC itself. For example, a performance gap of 20% on a classification dataset does not mean that AUROC is 20% lower than that of the best-performing model. Rather, it means that *reducing the current model's* $1-\mathrm{AUROC}$ *error by 20% would bring it to the best performance observed on that dataset*.

Reporting precision. Aggregate results are computed from full-precision scores. Reaggregating individual scores rounded to four decimal places may yield differences in the last digit relative to aggregates computed from the original full-precision values. Bold formatting is applied to the best displayed mean among the values compared in each table.

F Interpreting Residual Correction and Blended Distillation

We evaluate regression risk on a new sample $(X, Y) \sim P$ that is independent of the fixed training outputs. Let P_X denote the marginal distribution of X. We assume $\mathbb{E}[Y^2] < \infty$ and square integrability of all prediction functions considered. Define the regression function as $\mu(x) = \mathbb{E}[Y|X = x]$. For functions $f, h \in L_2(P_X)$, we write

$$\|f\|_2^2 = \mathbb{E}[f(X)^2],\ \langle f, h\rangle_2 = \mathbb{E}[f(X)h(X)]$$

For an X-measurable random variable $h(X)$, we also write $\|h(X)\|_2 = \|h\|_2$. For a general random variable $Z = Z(X, Y)$, we distinguish its norm as $\|Z\|_{L_2(P)} = \{\mathbb{E}[Z^2]\}^{1/2}$. We further define

$$R(f) = \mathrm{MSE}(f) = \mathbb{E}[(Y - f(X))^2],\ \mathrm{RMSE}(f) = \sqrt{R(f)}$$

The quantity Q_α denotes the blended-target objective, distinct from risk with respect to the original response variable. Function equalities are interpreted as holding P_X-almost everywhere where appropriate. In this section, G denotes an arbitrary fixed baseline predictor. In Eq. (2), $G = G_{h-1}$, with $G_0 = F_0$ at the first stage and $G_{h-1} = T_{h-1}$ at subsequent stages.

F.1 No New Information in Residual-State Representations

The conditional mutual information between two random variables is defined as $I(Y; Z|X) = \mathbb{E}_X\left[D_{\mathrm{KL}}\left(P_{Y,Z|X} \| P_{Y|X} \otimes P_{Z|X}\right)\right]$. We denote the σ-field generated by the random variable X by $\sigma(X) = \sigma\left(\{X^{-1}(A) : A \in \mathcal{B}_\mathcal{X}\}\right)$, where $\mathcal{B}_\mathcal{X}$ is the σ-field on the value space $\mathcal{X}$ of X. The notation $\sigma(X, Z)$ denotes the σ-field jointly generated by X and Z.

Proposition 1. *Fix the outputs of training and model selection, and let $Z = \hat{q}_r(X)$ be the residual-state representation for a new sample $(X, Y) \sim P$. Since Z is a measurable function of X,*

$$Y \perp Z|X,\ I(Y; Z|X) = 0.$$

In particular, for regression, $\mathbb{E}[Y|X, Z] = \mathbb{E}[Y|X] = \mu(X)$. Thus, under squared loss, the Bayes risk attainable by a predictor with an unrestricted function class is the same whether it uses only X or uses (X, Z).

Proof. Since $Z = \hat{q}_r(X)$ is a measurable function of X, $\sigma(Z) \subseteq \sigma(X)$. Hence,

$$\sigma(X, Z) = \sigma(X) \vee \sigma(Z) = \sigma(X),$$

where $\vee$ denotes the smallest σ-field generated by the two σ-fields. Therefore, once X is given, additionally conditioning on Z does not change the conditional distribution of Y: $P_{Y|X,Z} = P_{Y|X}$. It follows that

$$Y \perp Z|X,\ I(Y; Z|X) = 0.$$

We also have $\mathbb{E}[Y|X, Z] = \mathbb{E}[Y|X]$. □

F.2 Alignment with Residuals

Let the correction function in Eq. (2) be $d(x) = \gamma \hat{d}_h(x, \hat{q}_r(x))$.

Proposition 2 (Exact risk difference from residual correction)**.** *For a fixed predictor G and correction function d,*

$$\mathrm{MSE}(G + d) - \mathrm{MSE}(G) = -2\mathbb{E}[(Y - G(X))d(X)] + \mathbb{E}[d(X)^2].$$

Thus, the correction improves performance if and only if $2\mathbb{E}[(Y - G(X))d(X)] > \mathbb{E}[d(X)^2]$.

Proof. Expand $(Y - G(X) - d(X))^2 - (Y - G(X))^2 = -2(Y - G(X))d(X) + d(X)^2$ and take expectations. □

An additional representation alone does not guarantee improvement; the correction must align with the remaining error that is predictable from the inputs. If the baseline model equals the conditional mean μ, the residual has zero conditional mean, and no X-measurable correction can strictly improve the population squared risk. This does not require independence between the residual and the inputs.

F.3 Approximation Gains from Representation Augmentation in Restricted Correction Classes

Holding the baseline predictor fixed, we analyze how adding a residual-state representation as an input affects the best approximation error attainable within a restricted correction function class. The result concerns the expansion of the correction function class induced by a fixed representation, not the addition of new observed information.

Fixed representation and mean residual. Fix a correction stage and a fitting context, and omit stage and fold indices. Let G denote the baseline predictor, and represent the learned residual-state probability predictor as a measurable function

$$q_r : \mathcal{X} \longrightarrow \Delta^{K-1},\ Z = q_r(X)$$

where $\Delta^{K-1} = \{u \in \mathbb{R}^K_{\geq 0} : \sum_{k=1}^K u_k = 1\}$. All training outputs, including feature selection, preprocessing, and context construction, are held fixed, and their effects are incorporated into the function $q_r(x)$ evaluated on the original input x. The function q_r gives predicted state probabilities and is not assumed to equal the true conditional state probabilities. Neither the ground-truth label nor the actual residual of a new sample is used to compute $q_r(x)$.

For a new sample $(X, Y) \sim P$ independent of these training outputs, assume $\mathbb{E}[Y^2] < \infty$ and $G \in L_2(P_X)$. All expectations are computed with the training outputs fixed, and conditioning on them is omitted from the notation. Define the regression function $\mu(x) = \mathbb{E}[Y|X = x]$ and the mean residual function

$$d^\star_G(x) = \mu(x) - G(x) = \mathbb{E}[Y - G(X)|X = x]$$

The function $d^\star_G$ is the population-optimal correction when all square-integrable X-measurable correction functions are allowed and is distinct from the correction model actually learned.

Effective correction function classes. Let $\mathcal{G}_T^{(X)}$ and $\mathcal{G}_T^{(X,Z)}$ denote the function classes for the teacher's residual correction model using only the original inputs and using the residual-state representation as an additional input, respectively. Applying the same fixed shrinkage coefficient $\gamma \geq 0$ in both cases, define

$$\mathcal{D}_X = \left\{ x \mapsto \gamma d_T(x) : d_T \in \mathcal{G}_T^{(X)} \right\},$$
$$\mathcal{D}_{XZ} = \left\{ x \mapsto \gamma d_T\big(x, q_r(x)\big) : d_T \in \mathcal{G}_T^{(X,Z)} \right\}$$

Here, d_T denotes the correction model before shrinkage, while an element d of $\mathcal{D}_X$ or $\mathcal{D}_{XZ}$ is an effective correction function that incorporates shrinkage. Both classes consist of real-valued functions evaluated on the original input x and are assumed to be nonempty subsets of $L_2(P_X)$.

Assume that, under the same complexity constraints, the augmented correction model can also represent the original correction functions that do not use Z, so that

$$\mathcal{D}_X \subseteq \mathcal{D}_{XZ}$$

For example, this inclusion holds under the same limits on tree count and depth if splits along the original X coordinates remain available and the additional coordinates can be ignored.

Proposition 3 (Approximation gains from representation augmentation in restricted correction classes)**.** *Under the setup above, let*

$$A_X = \inf_{d \in \mathcal{D}_X} \|d^\star_G - d\|_2^2,\ A_{XZ} = \inf_{d \in \mathcal{D}_{XZ}} \|d^\star_G - d\|_2^2$$

Then $0 \leq A_{XZ} \leq A_X < \infty$*, and*

$$\inf_{d \in \mathcal{D}_X} \mathrm{MSE}(G + d) = \mathbb{E}[\mathrm{Var}(Y|X)] + A_X,\ \inf_{d \in \mathcal{D}_{XZ}} \mathrm{MSE}(G + d) = \mathbb{E}[\mathrm{Var}(Y|X)] + A_{XZ}.$$

Consequently,

$$\begin{aligned}\Delta_{\mathrm{rep}} &= A_X - A_{XZ} \\ &= \inf_{d \in \mathcal{D}_X} \mathrm{MSE}(G + d) - \inf_{d \in \mathcal{D}_{XZ}} \mathrm{MSE}(G + d) \geq 0\end{aligned}$$

is the population approximation gain obtained by adding the fixed residual-state representation. In particular, if $\Delta_{\mathrm{rep}} > 0$, then even though Z is a deterministic function of X, the risk infimum over the augmented correction class is strictly smaller than that obtained using only the original inputs. Neither infimum is assumed to be attained.

Proof. For any $d \in L_2(P_X)$,

$$Y - G(X) - d(X) = Y - \mu(X) + d_G^\star(X) - d(X).$$

Since $\mathbb{E}[Y - \mu(X)|X] = 0$ and $d_G^\star - d$ is X-measurable,

$$\mathbb{E}[(Y - \mu(X))(d_G^\star(X) - d(X))] = 0.$$

Therefore,

$$\begin{aligned}\mathrm{MSE}(G + d) &= \mathbb{E}[(Y - \mu(X))^2] + \|d_G^\star - d\|_2^2 \\ &= \mathbb{E}[\mathrm{Var}(Y|X)] + \|d_G^\star - d\|_2^2.\end{aligned}$$

Taking the infimum over each correction class yields the two risk decompositions. Moreover, since $\mathcal{D}_X \subseteq \mathcal{D}_{XZ}$,

$$A_{XZ} = \inf_{d \in \mathcal{D}_{XZ}} \|d_G^\star - d\|_2^2 \leq \inf_{d \in \mathcal{D}_X} \|d_G^\star - d\|_2^2 = A_X.$$

Both approximation errors are finite because $d_G^\star \in L_2(P_X)$ and the two correction classes are nonempty subsets of $L_2(P_X)$. Subtracting the two risk decompositions gives the identity for Δ_{rep}. □

Additional information versus representation augmentation. This result does not contradict Proposition 1. Since $Z = q_r(X)$ is a measurable function of X, it adds no new observed information and does not change the Bayes risk of an unrestricted predictor. For a restricted correction model, however, the additional coordinates can change the range of representable functions. If a mean residual structure that requires complex partitions in the original feature space can be approximated with simpler tree splits using the coordinates of $q_r(X)$, the approximation error can decrease even under the same complexity constraints. This does not guarantee $\Delta_{\mathrm{rep}} > 0$ for every state representation.

Relation to the learned teacher. For the teacher in the main text, the effective correction function is $d(x) = \gamma \hat{d}_h\big(x, q_r(x)\big)$. The proposition compares the best risks in two correction classes under the same G and fixed q_r and does not assume that the correction d actually learned attains the corresponding infimum. Furthermore, since the risk difference relative to the baseline predictor is

$$\mathrm{MSE}(G + d) - \mathrm{MSE}(G) = \|d_G^\star - d\|_2^2 - \|d_G^\star\|_2^2$$

the learned correction strictly improves G if and only if $\|d_G^\star - d\|_2 < \|d_G^\star\|_2$. The risk difference between learned teachers depends not only on the population approximation gain but also on finite-sample estimation, regularization, and optimization. Thus, $\Delta_{\mathrm{rep}} \geq 0$ alone guarantees neither improvement of the learned teacher nor better performance of the final student.

F.4 Population Objective of Blended Targets

We analyze the population objective corresponding to the student's actual learning procedure with OOF (out-of-fold) blended targets. Blending teacher predictions with observed responses does not generally make the teacher itself the student's population target. Under squared loss, the natural population target is an interpolation between the conditional mean and a weighted average of the fold-specific teachers.

After training and model selection, fix the selected stage $\hat{h}$. At this stage, let there be L folds, with the training-sample index set partitioned as $\{1, \ldots, n\} = \bigcup_{l=1}^{L} I_l$. The folds are pairwise disjoint, $n_l = |I_l| > 0$, and $l(i)$ is the fold index satisfying $i \in I_{l(i)}$. We omit the stage index in this section and denote the teacher trained on data excluding fold l by

$$T_l : \mathcal{X} \longrightarrow \mathbb{R},\ T_l(x) = T_{\hat{h}}^{(-l)}(x),\ l = 1, \ldots, L$$

Thus, each OOF prediction in the main text is $t_i^{\mathrm{OOF}} = T_{\hat{h}}^{(-l(i))}(x_i) = T_{l(i)}(x_i)$, and the OOF prediction vector at the stage selected in the algorithm is

$$\mathbf{t}_{\hat{h}}^{\mathrm{OOF}} = \left(t_1^{\mathrm{OOF}}, \ldots, t_n^{\mathrm{OOF}}\right)^\top$$

Let w_l be the weight assigned to each fold, with $w_l \geq 0$ and $\sum_{l=1}^{L} w_l = 1$. To match the implementation, one may use $w_l = n_l/n$, the fraction of all OOF samples belonging to each fold.

For the population analysis, the selected stage and training outputs are fixed, and $T_1, \ldots, T_L$ are treated as fixed functions. Although teachers are random functions estimated from training data, we condition on the training and model-selection results and consider population risk on a new sample $(X, Y) \sim P$ independent of the entire training and selection process.

Define the weighted mean of the fold-specific teachers as $\bar{T}_{\mathrm{CF}}(x) = \sum_{l=1}^{L} w_l T_l(x)$. This function averages the outputs of the fold-specific teachers at the same input x. By contrast, t_i^{OOF} is the prediction of the single teacher trained without the fold containing sample i and is generally not equal to $\bar{T}_{\mathrm{CF}}(x_i)$.

Using the regression function μ defined above and fixing the mixing weight $\alpha \in [0, 1]$, define

$$g_\alpha(x) = (1-\alpha)\mu(x) + \alpha \bar{T}_{\mathrm{CF}}(x).$$

For a candidate student function s, define the population blended-target risk as

$$Q_\alpha(s) = \sum_{l=1}^{L} w_l \mathbb{E}[\{(1-\alpha)Y + \alpha T_l(X) - s(X)\}^2]$$

This is the population counterpart of the squared loss for the blended target $(1-\alpha)Y + \alpha T_l(X)$ actually used in fold l, averaged according to the fold weights.

Lemma 1 (Orthogonal decomposition of the blended objective)**.** *For fixed $T_1, \ldots, T_L \in L_2(P_X)$, assume $\mathbb{E}[Y^2] < \infty$, $w_l \geq 0$, $\sum_{l=1}^{L} w_l = 1$, and $\alpha \in [0, 1]$. For any $s \in L_2(P_X)$,*

$$\begin{aligned} Q_\alpha(s) = (1-\alpha)^2 \mathbb{E}\left[\mathrm{Var}(Y|X)\right] + \alpha^2 \sum_{l=1}^{L} w_l \mathbb{E}\left[\left\{T_l(X) - \bar{T}_{\mathrm{CF}}(X)\right\}^2\right] \\ + \mathbb{E}\left[\left\{s(X) - g_\alpha(X)\right\}^2\right]. \end{aligned} \tag{3}$$

Thus, unconstrained minimization over all of $L_2(P_X)$ gives

$$s_\alpha^\star = g_\alpha \ P_X\text{-a.s.},$$

and, in general, the population-optimal function is $g_\alpha = (1-\alpha)\mu + \alpha \bar{T}_{\mathrm{CF}}$ rather than the teacher mean $\bar{T}_{\mathrm{CF}}$ itself.

Proof. By the definition of the conditional mean, we can write $Y = \mu(X) + \epsilon$, where

$$\mathbb{E}[\epsilon|X] = 0, \ \ \mathbb{E}[\epsilon^2|X] = \mathrm{Var}(Y|X). \tag{4}$$

In particular, for any square-integrable function $h(X)$ of X,

$$\mathbb{E}[\epsilon h(X)] = \mathbb{E}\left[h(X)\mathbb{E}[\epsilon|X]\right] = 0.$$

Thus, ϵ is orthogonal to all X-measurable functions in the $L_2(P)$ sense. For each fold, let $\Delta_l(x) = T_l(x) - \bar{T}_{\mathrm{CF}}(x)$ denote the teacher's deviation from the mean. By definition, for every x,

$$\sum_{l=1}^{L} w_l \Delta_l(x) = 0. \tag{5}$$

Also, writing $a_{\alpha,s}(x) = g_\alpha(x) - s(x)$ gives

$$\begin{aligned} & (1-\alpha)Y + \alpha T_l(X) - s(X) \\ = \ & (1-\alpha)\{\mu(X) + \epsilon\} + \alpha\{\bar{T}_{\mathrm{CF}}(X) + \Delta_l(X)\} - s(X) \\ = \ & (1-\alpha)\epsilon + a_{\alpha,s}(X) + \alpha\Delta_l(X). \end{aligned}$$

Expanding the square yields

$$\begin{aligned} & \{(1-\alpha)Y + \alpha T_l(X) - s(X)\}^2 \\ = \;& (1-\alpha)^2\epsilon^2 + a_{\alpha,s}(X)^2 + \alpha^2\Delta_l(X)^2 \\ + \;& 2(1-\alpha)\epsilon a_{\alpha,s}(X) + 2\alpha(1-\alpha)\epsilon\Delta_l(X) + 2\alpha a_{\alpha,s}(X)\Delta_l(X). \end{aligned}$$

The expectations of the first two cross terms are each zero by (4). Indeed, both $a_{\alpha,s}(X)$ and $\Delta_l(X)$ are functions of X, so $\mathbb{E}[\epsilon a_{\alpha,s}(X)] = 0$, $\mathbb{E}[\epsilon\Delta_l(X)] = 0$. For the last cross term, taking the weighted average over folds and using (5) gives

$$\sum_{l=1}^{L} w_l \mathbb{E}[a_{\alpha,s}(X)\Delta_l(X)] = \mathbb{E}[a_{\alpha,s}(X)\sum_{l=1}^{L} w_l\Delta_l(X)] = 0$$

It follows that $Q_\alpha(s) = (1-\alpha)^2\mathbb{E}[\epsilon^2] + \mathbb{E}[a_{\alpha,s}(X)^2] + \alpha^2\sum_{l=1}^{L} w_l\mathbb{E}[\Delta_l(X)^2]$. Combining $\mathbb{E}[\epsilon^2] = \mathbb{E}\left[\mathbb{E}[\epsilon^2|X]\right] = \mathbb{E}[\mathrm{Var}(Y|X)]$ with $\mathbb{E}[a_{\alpha,s}(X)^2] = \|s - g_\alpha\|_2^2$ and $\mathbb{E}[\Delta_l(X)^2] = \left\|T_l - \bar{T}_{\mathrm{CF}}\right\|_2^2$ yields Eq. (3). □

Interpreting the decomposition. Of the three terms in Eq. (3), the first, $(1-\alpha)^2\mathbb{E}[\mathrm{Var}(Y|X)]$, is the irreducible response noise introduced by including the observed response Y in the blended target. Since this term is independent of s, it does not change the population-optimal student function.

The second term, $\alpha^2\sum_{l=1}^{L} w_l \left\|T_l - \bar{T}_{\mathrm{CF}}\right\|_2^2$, is the cross-fitted teacher dispersion, measuring how much the fold-specific teachers differ. It is zero if every fold yields the same teacher function. Conversely, large variation in teachers across folds increases the variability of the blended targets themselves and raises the minimum attainable Q_α. This term is also independent of s. The final term, $\|s - g_\alpha\|_2^2$, directly affects student selection. Thus, at the population level, learning from blended targets can be interpreted as $L_2(P_X)$ approximation of g_α.

Interpolation between the teacher mean and the Bayes regression function. The function g_α can be rewritten in two forms:

$$\begin{aligned} g_\alpha &= \mu + \alpha(\bar{T}_{\mathrm{CF}} - \mu) \\ &= \bar{T}_{\mathrm{CF}} + (1-\alpha)(\mu - \bar{T}_{\mathrm{CF}}). \end{aligned}$$

Thus, g_α lies between the teacher mean and the Bayes regression function μ. In particular,

$$\|g_\alpha - \mu\|_2^2 = \alpha^2\left\|\bar{T}_{\mathrm{CF}} - \mu\right\|_2^2,\ \left\|g_\alpha - \bar{T}_{\mathrm{CF}}\right\|_2^2 = (1-\alpha)^2\left\|\mu - \bar{T}_{\mathrm{CF}}\right\|_2^2.$$

As the teacher weight α decreases, g_α approaches μ. When $\alpha = 1$, $g_1 = \bar{T}_{\mathrm{CF}}$ corresponds to pure distillation; when $\alpha = 0$, $g_0 = \mu$ recovers the population target of standard supervised regression. This result allows the blending of ground-truth labels and teacher predictions to be interpreted as shrinking the teacher's population error toward the true conditional mean.

Indeed, the standard squared prediction risk on a new sample, $R(s) = \mathbb{E}[(Y - s(X))^2]$, decomposes as $\mathbb{E}[\mathrm{Var}(Y|X)] + \|s - \mu\|_2^2$. Thus, assuming that g_α can be learned exactly without restrictions,

$$R(g_\alpha) = \mathbb{E}[\mathrm{Var}(Y|X)] + \alpha^2\left\|\bar{T}_{\mathrm{CF}} - \mu\right\|_2^2.$$

By contrast, the risk of the teacher mean is

$$R(\bar{T}_{\mathrm{CF}}) = \mathbb{E}[\mathrm{Var}(Y|X)] + \left\|\bar{T}_{\mathrm{CF}} - \mu\right\|_2^2.$$

The Bayes excess risk of g_α is α^2 times that of the teacher mean. This is a population comparison assuming exact realization of g_α, however, and does not guarantee improvement for a finite-sample student.

Restricted student function class. The actual student cannot use all functions in $L_2(P_X)$ and is instead trained within a restricted function class, such as that of XGBoost, denoted by $\mathcal{G}_S$. For the unregularized population objective, Lemma 1 gives

$$\arg\min_{s\in\mathcal{G}_S} Q_\alpha(s) = \arg\min_{s\in\mathcal{G}_S} \|s - g_\alpha\|_2^2.$$

When a minimizer exists, the restricted population optimum is a best approximation to g_α within $\mathcal{G}_S$. In general, it need not be a best approximation to T_{CF}. In particular, if $\mathcal{G}_S$ is a nonempty closed convex subset of $L_2(P_X)$, this can be interpreted as the L_2 projection of g_α.

The final performance of the actual student therefore depends on at least two factors: how favorably blending shifts the teacher mean to g_α, and how well the restricted function class $\mathcal{G}_S$ can approximate this new target. If the approximation error within the student function class is too large, the fact that g_α is closer to μ than T_{CF} does not guarantee an improvement for the learned student.