The Surprising Effectiveness of One-Step Boosting for Tabular Foundation Models

Anonymous authors
Paper under double-blind review

Abstract

Tabular foundation models (TFMs) predict from labeled examples without updating their parameters, yet those examples can also support a small task-specific correction. We adjust class scores with a validation-selected class-prior shift, then fit ridge regression to cross-fitted probability residuals and add its output to the scores when it improves validation macro-F1. Because the residuals are negative gradients of multiclass cross-entropy, this is one step of supervised boosting. On datasets with uncorrected validation macro-F1 below 0.60 (76/77/74 for TabPFN/TabICL/TabFM), the combined adjustments raise mean test macro-F1 by 6.93/6.50/8.21 points. The ridge step adds 2.41/3.29/5.16 points beyond the shift and improves one-vs-rest score-ranking AUC by 2.10/3.58/4.30 points. Even at validation macro-F1 of 0.60–0.80, it adds 1.44 macro-F1 and 0.89 ranking-AUC points on average. On a low-performing TabPFN subset, the ridge gain is +2.95 with the full labeled context and +5.19 with one quarter of it. On one A100, four-fold cross-fitting and ridge fitting take a median 11.65 s per TabPFN dataset beyond a shared 3.56 s base pass; once backbone scores exist, applying both corrections including test-row featurization takes a separately measured median 31 ms per 1,000 rows. Thus even capable TFMs benefit from a small, lightweight supervised output update.

1 Introduction

Tabular foundation models such as TabPFN (Hollmann et al., 2023; 2025), TabICL (Qu et al., 2025), and TabFM (Kong and Das, 2026) predict by conditioning on a labeled table, without optimizing pretrained parameters for each new task. Yet that context is itself a supervised dataset. A model may assign a minority class substantial probability across many rows while rarely selecting it as the top label. Moving the class's decision boundary can address that under-selection; identifying which rows belong to it requires a row-dependent adjustment. We ask how much of each improvement a small task-specific fit can recover, even when in-context prediction is already useful. Test-time training and fine-tuning adapt the backbone on target-task examples (Gozeten et al., 2025; Rubachev et al., 2025); we leave it frozen and learn directly from its prediction errors.

Our method makes two adjustments with different effects. A validation-selected class-prior shift moves decision boundaries through a single scalar, changing which class wins without changing the ordering of rows by a given class score. To learn a row-dependent adjustment, we score each training row without its own fold in the context and fit ridge regression to the difference between its label and the predicted class probabilities. We add the fitted residual to the frozen model's class scores. Because this residual is the negative cross-entropy gradient with respect to those scores, the procedure amounts to one regularized functional-gradient step. The two effects suggest different questions: are minority classes being selected too rarely despite receiving reasonable probability mass, and can simple features improve the ranking of rows within each class? Both can be tested without updating the TFM.

Across TabPFN, TabICL, and TabFM, the two adjustments improve test macro-F1 by 6.93, 6.50, and 8.21 points on datasets selected by uncorrected validation macro-F1 below 0.60. The ridge step also increases score-ranking AUC, isolating an improvement that cannot come

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.

from a class-wide shift. The result extends beyond the weakest cases: for validation macro-F1 in $[0.60, 0.80)$, the ridge step adds 1.44 macro-F1 points on average across the three models. Sampling labels from the models' own posteriors shows why a class decision adjustment can help even when those posteriors are correct by construction; fitting a much weaker task-only feature model reveals information that can complement an already capable backbone. We test both effects across model families and targeted interventions, with detailed controls in Appendices D–J. Together, these experiments show that a frozen in-context predictor can retain useful task-specific signal accessible through a single supervised output correction.

2 Related work

Tabular foundation models and task-specific adaptation. TabPFN, TabICL, and TabFM make predictions by conditioning on labeled examples (Hollmann et al., 2025; Qu et al., 2025; Kong and Das, 2026). Test-time training updates an in-context learner on these demonstrations (Gozeten et al., 2025); full fine-tuning of TabPFN adapts its parameters to the target dataset (Rubachev et al., 2025), while TuneTables optimizes the context (Feuer et al., 2024). We instead fit a small output correction using the labeled task data while keeping the backbone fixed. Appendix J uses fine-tuning to probe the source of the remaining gains; those experiments use partly different checkpoints and dataset populations.

Calibration and decision adjustment. Temperature scaling improves probability calibration but preserves argmax decisions and score ranking (Guo et al., 2017). A class-prior shift can improve macro-F1 by changing decisions even for a well-specified posterior, as in post-hoc logit adjustment and threshold selection (Menon et al., 2021; Lipton et al., 2014). We select its strength for validation macro-F1; label-shift approaches such as DistPFN instead address changing class proportions between training and test distributions (Lee, 2026). A gain from our shift therefore need not indicate a changed test prior. Appendix D compares these decision adjustments.

Stacking and boosting. Out-of-fold base-model predictions, familiar from stacked generalization (Wolpert, 1992), let us learn from training rows without allowing their labels into the context used to predict them. Fitting $y - p$ and adding the fitted response in score space follows the functional-gradient view of boosting under multinomial cross-entropy (Friedman, 2001). When the feature correction defines a task-only score model, score addition also has a log-linear pooling interpretation with an explicit class-prior term (Genest and Zidek, 1986; Hinton, 2002); averaging probabilities combines the models differently. These established operations let us test which class-decision and row-ranking gaps remain in frozen in-context predictors. Further connections to long-tailed learning and non-decomposable metrics appear in Appendix K.1.

3 Lightweight adaptation of a frozen TFM

Let $D_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^{n}$ be the labeled context for a C-class task. A frozen model returns $p_\theta(c \mid x, D)$, and we use class scores $z_c(x; D) = \log p_\theta(c \mid x, D)$. The backbone parameters θ do not change. Validation labels select correction hyperparameters; test labels are held out until evaluation.

Cross-fitted training scores. Split D_{tr} into K folds $F_1, \ldots, F_K$. For $i \in F_k$, evaluate $p_i = p_\theta(\cdot \mid x_i, D_{\mathrm{tr}} \setminus F_k)$. Thus each training target uses the prediction that the TFM makes when the target's label is absent from its context. This separation is necessary because an in-context learner can see the answer if a training row is scored with its own label among the demonstrations; fitting to such scores would understate the errors it makes on new rows. The implementation uses $K = 2$–4 (TabFM: 2); all final query predictions use the ordinary eligible training context. The exact context cap and folds appear in Appendix C.

Class-wide decision adjustment. Let $\pi_c = n_c/n$ denote a training class frequency. We optionally adjust each class score by

$$z_c^{(1)}(x) = z_c(x; D_{\text{tr}}) + \tau \log \pi_c, \tag{1}$$

where τ is selected on validation macro-F1. Negative τ favors rarer classes. The shift changes which class wins but adds the same value to every row's score for a given class, leaving rankings of the unnormalized scores unchanged. All ranking-AUC results use these class scores before cross-class normalization.

One-step ridge residual boosting. Apply the selected shift to the cross-fitted scores and write $p_{ic}^{(1)} = \text{softmax}(z_i^{(1)})_c$. Encode each row as $\widetilde{\phi}(x_i)$ using centered indicators for numerical ranges and categorical levels. Fit the multi-output ridge model

$$\widehat{W}_\lambda = \arg\min_W \sum_{i=1}^{n} \left\| \mathbf{e}_{y_i} - p_i^{(1)} - \widetilde{\phi}(x_i)^\top W \right\|_2^2 + \lambda \|W\|_F^2, \qquad z^{(2)}(x) = z^{(1)}(x) + g\, \widetilde{\phi}(x)^\top \widehat{W}_{\widehat{\lambda}}. \tag{2}$$

Here $p_i^{(1)}$ comes from the shifted out-of-fold score of training row i. We choose $\widehat{\lambda}$ on validation macro-F1 and set the dataset-level deployment indicator $g \in \{0, 1\}$ to one only if some ridge penalty strictly improves validation macro-F1 over the shift alone; otherwise $g = 0$ and the shifted scores are returned. Numerical features contribute indicators of training-derived ranges, and categorical features contribute level indicators; centering each indicator on the training split makes the fitted adjustment sensitive to a row's features rather than a freely fitted per-class intercept. For multiclass cross-entropy, $-\partial \ell / \partial z_{ic} = \mathbf{1}[y_i = c] - p_{ic}^{(1)}$, so the regression target is the negative functional gradient. Ridge regression approximates this gradient with squared error, and its fitted response supplies one regularized update to the scores. We call this procedure one-step residual boosting. There is no learned row-dependent gate or step size; a gated nonlinear corrector is evaluated separately in Appendix C.

Cost and deployment. Adapting to a task requires K TFM evaluations for cross-fitting, validation selection, feature construction, and ridge fitting. Each subsequent query batch requires one ordinary TFM evaluation followed by the score corrections; the ridge step first encodes new rows using the training-derived numerical ranges and categorical levels. On 76 low-performing TabPFN datasets, the median four-fold cross-fitting and ridge-fitting cost is 11.65 s per dataset, beyond a 3.56 s base pass shared by the two- and four-fold settings; using two folds costs 6.18 s with little measurable change in the correction gain (Table 4). Once backbone scores are cached, featurization and both corrections together take a median 30.66 ms per 1,000 test rows on these datasets (Appendix C.1).

4 Experiments and results

Experimental setup. We evaluate frozen TabPFN, TabICL, and TabFM alongside directly trained XGBoost and CatBoost (Chen and Guestrin, 2016; Prokhorenkova et al., 2018); additional comparisons include FT-Transformer, SAINT, and TP-BERTa. The datasets come from PMLB, TabArena, TabZilla, and TP-BERTa (Olson et al., 2017; Erickson et al., 2025; McElfresh et al., 2023; Yan et al., 2024). We use fixed train/validation/test splits and report test macro-F1 and one-vs-rest ranking AUC on class scores before cross-class normalization. For the primary analysis, each TFM's low-performing population is determined by its *uncorrected validation* macro-F1 < 0.60, before either correction is fitted. An 8K cap on total rows and the feature-encoding filters leave 76/77/74 datasets for TabPFN/TabICL/TabFM. Results over stronger validation-performance bands use the other eligible datasets rather than extrapolating from the selected low-performing set. Confidence intervals resample datasets; the comparisons use the same split within each model–dataset pair. Appendix C gives context limits, fold counts, software versions, filtering, selection grids, seeds, and population-specific scoring conventions.

4.1 Two complementary improvements

On each backbone's low-performing population, the combined method improves mean test macro-F1 by 6.93/6.50/8.21 points for TabPFN/TabICL/TabFM (Table 1). The class-prior shift contributes 4.53/3.21/3.05 points; the ridge step adds 2.41/3.29/5.16 beyond it. Before adaptation, the default XGBoost and CatBoost baselines lead the frozen TFMs by 1.6–5.3 macro-F1 points on the corresponding dataset sets. Applying the same corrections to the trees leaves the corrected TFMs ahead in mean macro-F1 on their respective populations, although only TabICL exceeds both trees after multiplicity correction (Table 15). The populations differ by backbone, so the means should be used to evaluate the correction within a model rather than to rank the three TFMs against each other. This gain comes from extracting information already present in the target task's labeled examples, with no change to the pretrained weights.

Table 1: Test macro-F1 ×100 on model-specific populations with uncorrected validation macro-F1 below 0.60 (one split; $n = 76/77/74$). The shift and ridge columns show cumulative mean performance; the AUC column shows the incremental ridge effect on unnormalized one-vs-rest class-score ranking, in percentage points. Tree baselines receive the same corrections and are evaluated on each TFM's population.

TFM	n	frozen	+ shift	+ ridge	ridge ΔAUC	corrected XGB / CB
TabPFN	76	46.26	50.78	53.19	+2.10	51.14 / 51.58
TabICL	77	48.24	51.45	54.74	+3.58	51.99 / 52.44
TabFM	74	46.36	49.41	54.57	+4.30	52.96 / 53.66

The two steps change different aspects of a prediction (Figure 1). Adding $\tau \log \pi_c$ moves the boundary at which class c wins, but cannot change the ordering of test rows by that class's unnormalized score. The ridge update depends on x and can improve that ordering: its one-vs-rest ranking-AUC gains are $+2.10/+3.58/+4.30$ points across the three TFMs. That distinction matters for interpreting the macro-F1 gains. A better decision rule can produce more correct labels without teaching the model which rows are better examples of a class; a ranking gain requires a row-dependent change. The appendix compares the shift with general per-class biases and DistPFN, and compares the ridge fit with nonlinear score and feature corrections (Appendices D and H.2). The restricted sequence captures much of the gain of a more flexible gated MLP benchmark, but the components are sequential increments rather than a unique additive decomposition.

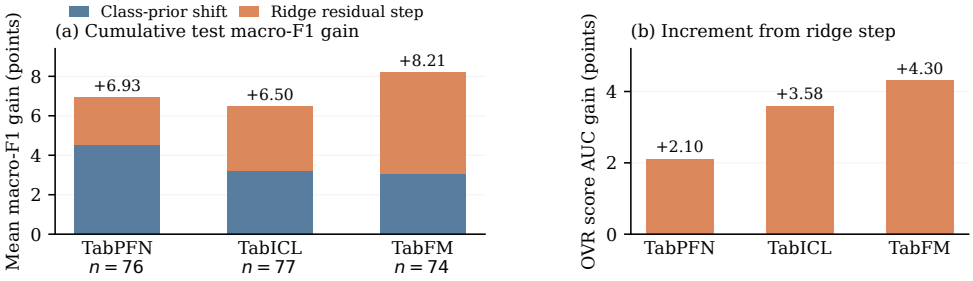

Figure 1: Lightweight adaptation on backbone-specific populations with uncorrected validation macro-F1 below 0.60. (a) Mean test macro-F1 gains from the class-prior decision shift and the incremental ridge residual step. (b) The ridge step also improves one-vs-rest ranking AUC measured on class scores before cross-class normalization; the class-wide shift has zero effect on this score ranking. Values are dataset means ($n = 76/77/74$ for macro-F1); TabICL's AUC bar uses 75 datasets with archived AUC rows. Uncertainty intervals and protocol details are in Appendix D.

How much flexibility does the shift need? The commonly used fixed correction $\tau = -1$ is one member of our one-scalar family and is exposed by TabPFN's `balance_probabilities` option. On the score-only populations it captures about 57/54/74% of the validation-tuned shift gain for TabPFN/TabICL/TabFM when always applied; only 27–31% of datasets select

a τ within 0.25 of -1 (Appendix E.2). The scalar is therefore doing useful task-specific work even though it changes scores identically across rows. Allowing an independent bias for every class does not improve on the single selected scalar in these comparisons, consistent with the limited validation data. This ablation helps separate the value of a class-wide decision adjustment from the value of tuning its strength to the task; it uses a score-only selection population distinct from Table 1.

The AUC improvement identifies a genuinely row-varying part of the update, but the ridge fit could also acquire a constant class offset on the test set: the indicators are centered on training rows, not on future rows. We check this by subtracting each class's test-set mean correction before scoring. On the separate $\leq$ 20K robustness populations, this re-centered update retains essentially the full macro-F1 increment: +2.27 versus +2.26 for TabPFN, +2.97 versus +3.14 for TabICL, and +4.92 versus +4.89 for TabFM (Appendix H.1). This test-time recentering is a diagnostic, not a deployable operation requiring access to an entire future test set; it shows that the observed gain is not merely a class-wide offset disguised as a feature model.

4.2 Class decisions can improve even with a correct posterior

Why does the class-prior shift help? On the score-only low-performing populations (84/83/79 datasets), the minority classes receive nearly the right *aggregate probability mass*: the mean predicted-to-observed mass ratio is 0.99 for each TFM. Yet their predicted labels occur at only 0.37/0.53/0.51 of their observed frequency before the shift (Table 16). The correction changes which class wins on a row, raising minority recall from roughly 20–27% to 40–42% while also raising minority precision from roughly 34–37% to 39–41%. It changes neither the original class-score ranking nor the evidence available on a given row. This is a statement about means across datasets, with appreciable dataset-level variation; the offset is selected for macro-F1 and worsens accuracy and proper scoring in these analyses. Figure 3 illustrates these two effects on decisions and row rankings.

A selection gain need not imply that the predicted probabilities are wrong. Macro-F1 can favor a decision rule different from argmax even if those probabilities are the true conditional probabilities (Lipton et al., 2014; Narasimhan et al., 2015). To measure this effect with the model's own scores, we draw independent synthetic validation and test labels from its predicted posterior, retune τ on the synthetic validation labels, and evaluate it against the synthetic test labels. The posterior is correct by construction in this experiment. Nevertheless, the shift gains +3.69/+2.26/+2.42 macro-F1 points for TabPFN/TabICL/TabFM (Table 21), against +4.53/ + 3.21/ + 3.05 on real labels. The observed-minus-synthetic intervals all include zero. This establishes that the decision rule and the *shape* of the model's posterior can generate a substantial gain without miscalibration; it does not establish that the real-data gain is caused entirely by the decision rule, since the synthetic-to-observed ratios have wide intervals. After selecting a calibrator on validation log-loss, the shift still recovers +4.52/ + 2.38/ + 2.73 points (Appendix E.5). Controls using accuracy under synthetic labels and sensitivity to injected class-score distortions further delimit this interpretation (Appendices E.3 and E.4).

Decision adjustment versus target-prior estimation. The frequency term in Eq. 1 does not assume that the test distribution has a different class prior from training. A label-shift estimator would try to infer that change; here τ is selected to improve a non-decomposable decision metric. We test this distinction in two ways. First, the class-prior shift gain remains +6.11 [+3.64, +8.69] macro-F1 points in the lowest quartile of training-to-test class-proportion divergence, pooled across the three TFMs; the quartile's median Jensen–Shannon divergence is approximately 4×10^{-4} nats (Appendix E.2). The divergence contrast is confounded with class count, so the inference is the presence of a gain even when the marginal shift is very small, not a causal effect of low divergence. Second, on the same features, an LDA model still gains +1.89 [+0.49, +3.33] points after its empirical prior is replaced with a uniform prior, compared with +4.18 [+2.71, +5.72] under its empirical prior (Appendix K). These controls distinguish choosing decisions for macro-F1 from simply substituting an estimated target prior; label-shift estimators are evaluated separately in Appendix E.2.

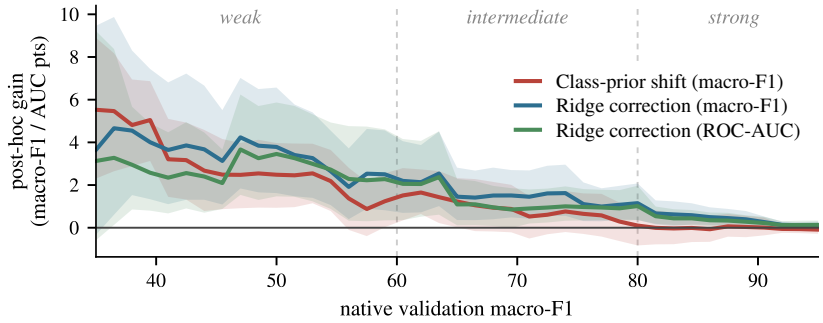

Figure 2: Gains from the class-prior shift and ridge residual correction across validation-performance bands (three models pooled; 95% dataset-bootstrap bands).

4.3 The residual step remains useful for stronger models

The row-dependent gain extends beyond the datasets selected for low validation macro-F1. Over *all eligible* model–dataset pairs with uncorrected validation macro-F1 in $[0.60, 0.80)$, the class-prior shift contributes $+0.66\,[-0.41, +1.75]$ points, while the ridge step still adds $+1.44\,[+0.47, +2.60]$ macro-F1 and $+0.89\,[+0.10, +1.81]$ ranking-AUC points across the three TFMs (Figure 2). Here the second increment matters more than the first: even a model that already makes reasonably good predictions can benefit from a small task-specific fit. At validation macro-F1 ≥ 0.80, the mean shift increment is near zero and the ridge increment is small ($+0.23\,[+0.05, +0.47]$ macro-F1), so the useful regime is broad but not unlimited. Appendix H reports counts and backbone-level estimates; TabFM contracts faster than TabPFN and TabICL in the intermediate band.

Less labeled context leaves more recoverable signal. We repeat the TabPFN v3 inference and the complete cross-fitting procedure with stratified subsets comprising 100%, 50%, and 25% of each original training split, holding validation and test rows fixed (Table 2). On the 63 low-performing datasets with at least 200 training rows, native macro-F1 falls from 44.14 to 40.80 as the context shrinks to one quarter (about 125 rows at the median). At the same time, the shift gain rises from +4.39 to +5.33 points and the incremental ridge gain from +2.95 to +5.19; on all 264 eligible datasets, the corresponding gains rise from $+1.07/+1.05$ to $+1.84/+1.76$. These are level estimates with dataset-bootstrap intervals, not paired confidence intervals for the change across fractions. The corrected gains therefore persist under a substantial reduction in labeled context, with larger point estimates where the native model has less context. The fractions also reduce the rows available for fitting the corrector, so the experiment varies both in-context information and supervised training size. An earlier context intervention relative to refitted trees asks a different question and does not show that the class-selection gap relative to trees increases as context shrinks (Appendix E.1.1).

Table 2: **Corrections with smaller labeled contexts on TabPFN v3.** Fraction f of the original training split is available both to the backbone and the freshly cross-fitted correction; validation and test splits stay fixed. Test macro-F1 $\times 100$ and incremental gains in points (dataset-bootstrap 95% intervals); the shift is measured against native, ridge against shifted scores, and ranking AUC is the ridge increment. The 63 low-performing datasets are a subset of the 264 datasets with ≥ 200 training rows, distinct from Table 1's 76-dataset population. One dataset with a single-class test split is excluded throughout.

f	native F1	shift gain	ridge gain	ridge ΔAUC
Low-performing subset ($n = 63$)				
1.00	44.14	$+4.39\,[+2.65, +6.26]$	$+2.95\,[+0.85, +5.59]$	$+2.08\,[+0.21, +4.47]$
0.50	42.19	$+5.50\,[+3.67, +7.47]$	$+4.72\,[+1.74, +8.40]$	$+1.93\,[+0.08, +4.14]$
0.25	40.80	$+5.33\,[+3.52, +7.25]$	$+5.19\,[+2.14, +8.62]$	$+3.34\,[+1.21, +5.85]$
All eligible datasets ($n = 264$)				
1.00	75.83	$+1.07\,[+0.48, +1.69]$	$+1.05\,[+0.46, +1.76]$	$+0.65\,[+0.16, +1.28]$
0.50	74.21	$+1.63\,[+0.94, +2.34]$	$+1.68\,[+0.86, +2.64]$	$+0.65\,[+0.15, +1.24]$
0.25	72.58	$+1.84\,[+1.12, +2.58]$	$+1.76\,[+0.93, +2.71]$	$+1.08\,[+0.52, +1.78]$

The additional information need not be learned from the TFM's residuals specifically. We fit the same centered feature indicators to $\mathbf{e}_y - \pi$, without observing any TFM predictions during fitting, and combine the resulting task-only feature model with the frozen predictor. Across eight model families on their low-performing populations, this diagnostic recovers much of the model-residual ridge gain (Table 32). The intermediate band is more revealing: the standalone feature model scores 51.6 macro-F1 against 68.8 for the frozen TFM, yet adding its correction after the class-prior shift improves the TFM by $+2.13\,[+0.63, +3.74]$ points (Table 3). The feature model is a weaker predictor in isolation but supplies information complementary to the TFM's scores.

Table 3: **Complementarity in the intermediate band** (0.60–0.80 validation macro-F1; 226 pairs over 99 datasets, three frozen models pooled; test macro-F1, dataset-clustered 95% intervals).

condition	frozen alone	task-only model alone	probability average	tuned log-linear pool	class-prior shift + task-only model
macro-F1	68.8	51.6	68.3	–	71.0
change	–	–	$-0.53\,[-2.09, +1.19]$	$+2.32\,[+1.08, +3.88]$	$+2.13\,[+0.63, +3.74]$

How the scores are combined is consequential. A validation-tuned *probability average* of the two models changes macro-F1 by $-0.53\,[-2.09, +1.19]$ in that intermediate band, whereas a tuned *log-linear pool* gains $+2.32\,[+1.08, +3.88]$, comparable to adding the feature correction in score space. The pooled score addition can itself be expressed as a log-linear combination of the frozen and task-only model scores (Appendix H). Thus the gain is evidence of complementary feature information, while the particular one-step residual fit is a simple way to access it. The task-only arm tests feature information with its own validation choices; the residual-fitting arm instead learns from the frozen model's cross-fitted errors. The encoded-state probe cannot establish where in the pretrained backbone this information is retained or lost (Appendix I).

Shared row-correction signal. The fitted adjustments are also partly transferable across predictors. Evaluating centered corrections on the same test rows, the three frozen TFMs have an excess cosine similarity of $+0.74\,[+0.63, +0.84]$ over random and permutation controls, while TFM–tree pairs retain an excess similarity of $+0.42\,[+0.23, +0.60]$ (Appendix H.3). A correction fitted on one model recovers 50–81% of another model's own row-dependent macro-F1 gain on the tested shared datasets. Within the intermediate validation-performance band, transferring a correction between frozen TFMs contributes $+1.15\,[+0.21, +2.33]$ points, compared with $+1.48\,[+0.48, +2.70]$ for a correction fitted on the target model (Appendix H). This overlap is consistent with simple feature ranges capturing task-level structure that several predictors leave available at their outputs. It does not imply that their internal representations or training mechanisms are the same.

4.4 Model-family comparisons and targeted controls

The class-wide and row-dependent gains have different patterns across model families (Figure 5). On each family's own low-performing datasets, TabPFN, TabICL, and TabFM show shift gains of $+4.53/+3.21/+3.05$ points, while XGBoost and CatBoost show $-0.32/+0.41$ (Table 24). The ridge correction helps beyond the shift in both pretrained and directly trained models, so its usefulness is not unique to in-context prediction. For example, the from-scratch FT-Transformer shows a $+4.07\,[+1.61, +6.96]$ ridge increment alongside a $+0.49$ class-prior shift gain on its own low-performing set. Conversely, fine-tuned TP-BERTa shows large gains in both channels, underscoring that this comparison concerns model families rather than a simple frozen-versus-trained dichotomy. To address differing dataset sets, a comparison restricted to datasets where both model groups are low-performing estimates $+1.97\,[+1.13, +2.81]$ more class-prior shift gain in the pretrained group with dataset fixed effects (Appendix F). This is an association across the evaluated families; architecture, objective, and adaptation procedure also differ. In particular, the TabICL and TabFM shift estimates vary with alternate splits (Appendix G).

The gains also persist with a newer backbone: on the *same* 76 datasets selected using TabPFN v3, TabPFN-3.5 starts $+2.28\,[+0.44, +4.08]$ macro-F1 points stronger, yet still gains $+3.06\,[+1.49, +4.62]$ from the class-prior shift and a further $+2.94\,[+0.83, +5.48]$ from ridge (Appendix F.4). Repeating the decomposition with a reselected v3.5 population gives the same qualitative pattern. Changing the dataset-size cap from 8K to 20K also retains both gains; across alternate splits, the ridge increment is more stable than the shift (Appendix G). These checks support the value of the correction without making the low-performing gate itself a universal deployment rule.

Two targeted interventions probe the distinction between the gains (Table 36). On the version-matched TabPFN fine-tuning population, class-balanced training reduces the remaining class-selection gain by $4.00\,[1.77, 6.32]$ points, while the change in ranking-AUC gain is $+0.11\,[-0.55, +0.77]$. Conversely, a centered feature head added to the output scores reduces the remaining row-ranking gain by $3.55\,[1.70, 5.87]$ AUC points over a pooled three-backbone population, while the class-selection change $-0.59\,[-1.61, +0.35]$ is unresolved. The head is an output-level bypass closely related to our ridge correction, so its effect on row ranking follows the intervention's design; the comparison does not locate an internal failure mode. These are diagnostics on distinct populations and, for fine-tuning, partly different checkpoints, not a matched performance or runtime comparison with full-model adaptation (Appendix J).

Computation. On the 76 low-performing TabPFN datasets, the median baseline pass for loading the context and scoring validation and test rows costs 3.56 s per dataset on one A100. Obtaining out-of-fold training residuals and solving the ridge problem costs a further 11.65 s with four folds, versus 6.18 s with two; fitting the class-prior shift takes 0.052 s from the existing validation scores (Table 4). Moving from four to two folds changes the recovered gain by $+0.03\,[-0.19, +0.24]$ macro-F1 and $-0.17\,[-0.49, +0.01]$ ranking-AUC points. Once backbone scores exist, the median cost of encoding 1,000 new rows and applying both corrections is 30.66 ms on these datasets, chiefly from the 30.18 ms feature encoding (Appendix C.1). For scale, separate A100 backbone measurements range from 238 to 1,177 ms per 1,000 rows across TabICL, TabPFN, and TabFM; extending the corrector timing to other backbones is an estimate, since their scoring was measured on a different dataset sample. The cross-fitted forward passes dominate additional fitting cost, while feature construction dominates correction cost at prediction time.

Table 4: **Fitting and applying the corrections on TabPFN v3.** Median seconds per dataset and milliseconds per 1,000 test rows on the 76 primary low-performing datasets (median 502/84/82 train/validation/test rows). Cross-fitting uses one A100 and includes out-of-fold backbone predictions; its two fit rows exclude a shared 3.56 s context-loading and validation/test-scoring pass. The 844 ms backbone query comes from a separate A100 benchmark; the correction-only prediction times use cached scores and exclude test-row featurization. The shift fit uses existing validation scores. The complete cached-score correction, including featurization, is measured separately in Appendix C.1.

operation	fit (s/dataset)	predict (ms/1,000 rows)
TabPFN v3 backbone setup / query	0.56	844
class-prior shift	0.052	0.105
ridge solve (cross-fitted scores cached)	0.010	0.296
ridge with four-fold cross-fitting	11.65	0.296
ridge with two-fold cross-fitting	6.18	0.296

5 Discussion and limitations

The two adjustments serve different objectives. The class-prior shift selects decisions for macro-F1; on the score-only populations it lowers accuracy by roughly 1.8–2.2 points and increases log-loss and Brier score (Appendix E). Its shifted scores should therefore not be treated as better probability forecasts. If calibrated probabilities are needed, the frozen model's original probabilities and the macro-F1 decision rule can be kept separate. The ridge adjustment changes both decisions and score ranking. Its gain is measured *after* the shift, so the component sizes depend on the order of fitting; they are useful for isolating information channels, not an additive attribution of an intrinsic model error (Appendix D).

The 6.5–8.2-point total gains describe model–dataset pairs selected by uncorrected validation macro-F1 below 0.60. The intermediate band provides a separate test of how far the row-dependent gain persists, and the effect becomes small once validation macro-F1 exceeds 0.80. Heterogeneity also matters within the selected set: the shift lowers test macro-F1 on 21–30% of datasets and the row correction on 20–22% (Appendix D). The ridge deployment indicator requires a strict validation gain over the shift, yet validation selection cannot prevent every test-set loss. The class-wide gain is stronger on binary tasks, where the shift is a one-parameter threshold adjustment; row-ranking gains have less consistent dependence on class count (Appendix G).

Two-fold cross-fitting lowers the median additional fitting time from 11.65 to 6.18 s without a detectable change in gain. The context-fraction experiment changes both the backbone's input and the corrector's training set, so it does not isolate either mechanism.

The synthetic-label control establishes a possible decision-rule gain under a correct posterior without attributing the real-data gain to that mechanism. Cross-family patterns do not isolate pretraining, and our internal-state probe cannot localize the row-dependent information (Appendix I).

6 Conclusion

Across frozen TabPFN, TabICL, and TabFM, a class-prior shift followed by one cross-fitted ridge step raises mean macro-F1 by 6.93, 6.50, and 8.21 points on low-performing tasks without updating pretrained parameters. The shift improves class decisions even with a correct synthetic-label posterior; the ridge step improves row ranking and remains useful at validation macro-F1 of 0.60–0.80. Both gains persist when TabPFN's labeled context shrinks. Task-only feature models show that simple feature signal complements a capable backbone; cross-family comparisons and targeted interventions distinguish class-selection and row-ranking effects. The out-of-fold backbone passes dominate adaptation; encoding new rows and applying both corrections takes a separately measured median 31 ms per 1,000 rows. This work provides evidence that even strong in-context tabular foundation models do not fully exploit the predictive information in their labeled context, and that a single residual boosting step can use that data to efficiently improve performance.

Reproducibility statement

Section 3 presents the mathematical formulation, cross-fitting procedure, residual fitting objective, and validation-based selection rule. Appendix C documents model and software versions, dataset selection and preprocessing, data splits, feature construction, hyperparameter grids, random seeds, and uncertainty estimation. Table 10 specifies the dataset populations and scoring conventions used in each analysis, and Appendix C.1 describes the timing protocols and hardware.

AI use statement

Generative AI tools assisted with manuscript language editing, figure preparation, and experiment implementation. The authors reviewed the AI-assisted material and take responsibility for the final manuscript, including its methods, results, citations, and conclusions.

References

Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 222–232, 2020.

Eugène Berta, Francis Bach, and Michael Jordan. Classifier calibration with ROC-regularized isotonic regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

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

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. An exact algorithm for F-measure maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

Krzysztof Dembczyński, Wojciech Kotłowski, Oluwasanmi Koyejo, and Nagarajan Natarajan. Consistency analysis for binary classification revisited. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 961–969, 2017.

James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 194–202, 1995.

Charles Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1697e3fb412da11dc9488249f9e7bbc9-Abstract-Datasets_and_Benchmarks_Track.html.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1022–1029, 1993.

Benjamin Feuer, Robin Tibor Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. TuneTables: Context optimization for scalable prior-data fitted networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Christian Genest and James V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Halil Alperen Gozeten, Muhammed Emrullah Ildiz, Xuechen Zhang, Mahdi Soltanolkotabi, Marco Mondelli, and Samet Oymak. Test-time training provably improves transformers as in-context learners. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 20266–20295. PMLR, 2025. URL https://proceedings.mlr.press/v267/gozeten25a.html.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

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

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

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Martin Jansche. Maximum expected F-measure training of logistic regression models. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 692–699, 2005.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020.

Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. CARTE: Pretraining and transfer for tabular learning. In *International Conference on Machine Learning (ICML)*, 2024.

Weihao Kong and Abhimanyu Das. Introducing TabFM: A zero-shot foundation model for tabular data. Google Research blog, 2026. URL https://research.google/blog/introducing-tabfm-a-zero-shot-foundation-model-for-tabular-data/. Code: https://github.com/google-research/tabfm.

Oluwasanmi O. Koyejo, Nagarajan Natarajan, Pradeep K. Ravikumar, and Inderjit S. Dhillon. Consistent binary classification with generalized performance metrics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Seunghan Lee. Mitigating label shift in tabular in-context learning via test-time posterior adjustment. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2605.04363.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pages 225–239, 2014.

Zachary C. Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 3122–3130, 2018.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2012.

Samuel McDowell, Nathan Stromberg, and Lalitha Sankar. Correcting class imbalance in prior-data fitted networks for tabular classification. *arXiv preprint arXiv:2605.21742*, 2026.

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

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*, 2021.

Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Harikrishna Narasimhan, Harish G. Ramaswamy, Aadirupa Saha, and Shivani Agarwal. Consistent multiclass algorithms for complex performance measures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2398–2407, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 625–632, 2005.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. InterpretML: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. PMLB: A large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(1):36, 2017.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large-Margin Classifiers*, pages 61–74. MIT Press, 2000.

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

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

Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Harish G. Ramaswamy, Harikrishna Narasimhan, and Shivani Agarwal. Consistent classification algorithms for multi-class non-decomposable performance metrics. *arXiv preprint arXiv:1501.00287*, 2015.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1): 21–41, 2002.

Zhanliang Wang, Hongzhuo Chen, Quan Minh Nguyen, Mian Umair Ahsan, and Kai Wang. Beyond logit adjustment: A residual decomposition framework for long-tailed reranking. In *Conference on Language Modeling (COLM)*, 2026.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Z. Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making pre-trained language models great on tabular prediction. In *International Conference on Learning Representations (ICLR)*, 2024.

Nan Ye, Kian Ming A. Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing F-measures: A tale of two approaches. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

The following experiments examine class-wide decision adjustment and row-dependent residual correction across datasets, model families, and training interventions. Each analysis states its dataset population and evaluation protocol; Table 10 collects these choices in one place.

A Additional TabPFN-3.5 checkpoint results

Table 5: **Both corrections remain effective on TabPFN-3.5.** We compare checkpoints on the same 76 datasets selected using TabPFN v3 validation predictions (macro-F1 ×100). Ridge gains are incremental over the class-prior shift. The released per-class threshold tuner precedes both corrections in the final row. Intervals and results on a population reselected using v3.5 are in Table 26.

checkpoint and decision rule	initial macro-F1	shift gain	ridge gain
TabPFN v3	46.26	+4.53	+2.41
TabPFN-3.5	48.53	+3.06	+2.94
TabPFN-3.5 + decision-threshold tuner	46.41	+4.26	+3.89

B A worked example of class-wide and row-dependent corrections

The class-wide adjustment changes how easily a rare class wins; the row-specific correction changes which rows the model regards as most like that class. Consider three classes A, B and a rare class R, and four rows with two numerical features (Table 6). Class R never wins, although the model already ranks the rows by their R score: Row 3 highest, then Rows 1 and 4 (tied), then Row 2.

Table 6: Uncorrected scores. R never has the largest score.

Row	Age	Salary	scores $[A, B, R]$	pred
1	27	45k	$[1.40, 1.10, 0.80]$	A
2	29	95k	$[1.20, 1.00, 0.70]$	A
3	52	45k	$[1.00, 1.10, 0.90]$	B
4	55	100k	$[0.90, 1.00, 0.80]$	B

Table 7: A class-wide adjustment (+0.35 to every R score) makes R more competitive but leaves the row ordering by R unchanged.

Row	uncorrected R	shifted R	rank
3	0.90	1.25	1
1	0.80	1.15	2 (tie)
4	0.80	1.15	2 (tie)
2	0.70	1.05	4

Class-wide adjustment. Adding the same constant +0.35 to the R score of every row (Table 7) makes R win against A and B on more rows, but because the adjustment is identical for all rows it cannot change their ordering by the R score: Rows 1 and 4 remain tied. This is exactly the invariance of Fact 1.

Row-specific correction. A correction that reads feature ranges can move rows differently. Give the R score an adjustment of +0.20 for Age 20–35 and −0.20 for Age 36–60, and +0.15 for Salary below 60k and −0.15 at or above it. The two tied rows then separate (Table 8): Row 1 (young, lower salary) rises and Row 4 (older, higher salary) falls. In the actual ridge residual correction the interval indicators are centered by their training frequencies, removing their average class-wide contribution, and one weight per class and feature range is learned jointly by ridge regression on out-of-fold training errors.

Why perfect calibration does not imply perfect macro-F1. Suppose 1,000 rows all receive $P(A) = 0.60$, $P(B) = 0.30$, $P(C) = 0.10$, and labels are sampled from them, giving about 600 rows of A, 300 of B and 100 of C. Argmax predicts A everywhere, so precision for A is about 0.60 at recall 1, giving $\mathrm{F1}_A \approx 0.75$ while $\mathrm{F1}_B = \mathrm{F1}_C = 0$ and macro-F1 is about 0.25. The model is perfectly calibrated here, the labels coming from its own probabilities, yet macro-F1 is poor because argmax always selects A — the intuition behind the self-consistency null of Section 4.2.

Table 8: After the class-wide adjustment, the feature-range terms move the tied rows in opposite directions.

Row	R after adj.	Age term	Salary term	final R
1	1.15	+0.20	+0.15	1.50
2	1.05	+0.20	−0.15	1.10
3	1.25	−0.20	+0.15	1.20
4	1.15	−0.20	−0.15	0.80

The ranking-preservation boundary (Section 3). For completeness, we state the standard ROC property used by our identification strategy; it is not a contribution.

Fact 1 (standard ROC invariance)**.** *Fix a class c and let $s_c(x)$ be its score for row x. If the same strictly increasing function g_c is applied to every row, $s_c(x) \mapsto g_c(s_c(x))$, then the one-vs-rest AUC for class c is unchanged. The set of rows on which class c becomes the final prediction may still change.*

This is the standard fact that ROC AUC depends only on score order (Fawcett, 2006); order-preserving constraints of this kind are also used in multiclass calibration (Rahimi et al., 2020; Berta et al., 2024). We compute ranking AUC on the class scores *before* cross-class normalization, where Fact 1 applies exactly. Normalization can in principle introduce small row-dependent changes through its normalizer, but empirically that effect is centered near zero (median change 0.00 AUC points on every backbone, pooled mean at most 0.26 points in magnitude), so it cannot account for the positive 2.10–4.30-point AUC gains of the row-dependent correction. The last paragraph of Section 4.1 adds an empirical check using existing posterior-adjustment methods.

These numbers are constructed for explanation, not taken from a benchmark dataset; Appendix D gives the population-level version of the same contrast.

C Implementation and evaluation details

Each dataset has a fixed training, validation, and test split. We select the low-performing subset for each model using its uncorrected validation predictions before fitting either correction. Cross-fitting requires additional backbone evaluations on training folds; validation and test predictions use the eligible training context.

Table 9: **The three corrections and their roles.** The flexible recoverability benchmark sizes the recoverable gap; the two restricted corrections identify whether that recovery comes from class selection or row ordering.

Correction	Reads	Form / output	Reorders rows?	Role
Flexible recoverability benchmark	model class scores + row feature-range indicators	MLP → row-dependent adjustment $\Delta z(x)$, gated by $a(x)$	Yes	Sizes the recoverable gap (benchmark, not attribution)
class-prior shift	training class frequencies only	one validation-tuned scalar; same class offset on every row	No	Class-selection component
ridge residual correction	centered row feature-range indicators	linear ridge on the post-class-prior shift per-class error → row-dependent adjustments	Yes	Row-ordering component

Fitted elements. Table 9 specifies the two corrections and a flexible recoverability benchmark. The flexible recoverability benchmark applies its adjustment through a learned row weight, $z^{\text{corr}}(x) = z^{\text{model}}(x) + a(x)\,\Delta z(x)$, with a validation safeguard that discards it when it does not improve validation macro-F1; the ridge residual correction is fitted on top of the class-prior shift and is deployed only when its best validation candidate strictly improves macro-F1 over the shift. Controls include arbitrary per-class biases, nonlinear score and feature corrections, the row weight and the safeguard. Every variant is evaluated on the same fixed dataset set per model.

Populations and reproducibility. Outside the fine-tuning audits, the alternate-split replication (Appendix G) and the context-size experiments (Table 2, Appendix E.1.1), every analysis uses one fixed train/validation/test split per dataset and a single seed-42 frozen-model inference; seeds $\{42, 43, 44, 45, 46\}$ vary only the MLP-based elements and the directly trained baselines. The low-performing condition, validation macro-F1 < 0.60, is set separately for each model from its uncorrected validation predictions, before any correction is fitted.

Primary decomposition population. We exclude datasets with more than 8K total rows, datasets for which no numerical feature yields a usable discretization, and datasets whose feature-range representation would exceed 4000 indicator columns. The resulting primary $\leq$8K populations contain 76, 77 and 74 datasets for TabPFN, TabICL and TabFM; the uncapped low-performing populations contain 84, 83 and 79.

Analysis-specific populations. Other analyses use the largest population satisfying their own requirements: the minority-selection analysis retains the score-only populations (84/83/79) and the fine-tuning audit uses the fine-tuning-eligible sets ($n = 71$ for TabPFN, which requires a minimum per-class count, and $n = 76$ for TabICL). Analyses marked "recomp." in Table 10 recompute the < 0.60 gate from their own uniformly scored cached validation probabilities, so datasets close to the threshold can differ; the feature-head population is $n = 75/76/85$ (99 pooled datasets). These filters are conservative: on the uncapped populations the class-prior shift gains remain $+4.86/ + 3.54/ + 3.33$, and the datasets excluded by the featurization requirement have *still larger* average gains ($+10.3$ and $+8.8$ for TabPFN and TabICL).

Feature representation and fitting detail. Numerical features are median-imputed and quantile-binned, the number of bins set by training-only entropy-based discretization (Fayyad and Irani, 1993) with label-free training quantiles as boundaries; categorical features contribute one indicator per training-observed level. The ridge residual correction fits ridge regression from the centered interval indicators to the post-shift per-class residual $\mathbf{1}[y = c] - \mathrm{softmax}(z^{(1)}(x))_c$, adds $\widetilde{\phi}(x)W$ to $z^{(1)}$ when selected by its dataset-level validation gate, and selects $\lambda \in \{0.3, 1, 3, 10, 30\}$ on validation. The flexible recoverability benchmark and the nonlinear feature-range control instead produce $\Delta z(x)$ with a two-hidden-layer, width-256 ReLU MLP (dropout 0.3, weight decay 10^{-4}, Adam at 3×10^{-3}, up to 300 epochs, early stopping), gated by a $2-16-1$ sigmoid MLP on the standardized maximum probability and entropy. The ridge residual corrector minimizes squared error with an L_2 penalty; the MLP-based correctors use cross-entropy, and the class-prior shift is selected by validation macro-F1. The training-row predictions are stratified and out of fold, with $k = \max(2, \min(4, \text{rarest training-class count}))$ folds for TabPFN and TabICL ($k = 4$ on 358 of 371 TabPFN datasets; TabICL's per-fold training posteriors use a context drawn only from that fold's training rows) and $k = 2$ for TabFM. The class-prior shift searches τ on a 41-point grid over $[-2, 2]$ (negative τ raises rarer classes since $\log \pi_c < 0$); the general per-class bias uses coordinate ascent on a 33-point grid over $[-4, 4]$.

Fine-tuning and baselines. Fine-tuning uses FinetunedTabPFNClassifier for up to 20 epochs (50 for the aggressive learning-rate/weight-decay grid), early-stopped at learning rate 10^{-5} (aggressive grid: learning rate $\{1, 2, 4\} \times 10^{-5}$, weight decay $\{10^{-3}, 10^{-2}, 10^{-1}\}$). Class-balanced fine-tuning oversamples every non-majority class to the majority-class count before the same cross-entropy step; output-layer-only fine-tuning freezes the encoder and updates 77,962 of 10,718,218 parameters (0.7%). XGBoost and CatBoost use library defaults and the FT-Transformer a four-trial Optuna search; seeds 43–46 fix the framework RNG, whereas the seed-42 FT-Transformer seeded only the Optuna sampler.

Model versions and inference configuration. The frozen TabPFN backbone is TabPFN v3, called through the public `tabpfn` package (version 8.0.3; `torch` 2.12.0, `scikit-learn` 1.7.2, `numpy` 2.2.6) with its library-default inference ensemble (8 members, softmax temperature 0.9, no override). Its `balance_probabilities` option, the $\tau = -1$ member of the class-prior shift family (Section 3), is left at its default `False`, so the component we report

on TabPFN is the one that flag, left off, leaves in place (enabled, it would recover about 0.57 of the tuned gain, Table 19); TabICL and TabFM expose no analogous flag. The fine-tuning audits instantiate the smaller TabPFN v2.5 through `FinetunedTabPFNClassifier`, with the version-matched frozen v2.5 baseline described next. TabICL is `tabicl` 2.1.1 with the `tabicl-classifier-v2-20260212` checkpoint at library defaults (8 members, softmax temperature 0.9, logit averaging; three hill-valley units averaged 4 members in the cross-fitted extraction of Appendix I); its fine-tuning uses `FinetunedTabICLClassifier` with the same checkpoint, whose no-update condition reproduces the frozen predictions bit-exactly. TabFM is the public `google/tabfm-1.0.0-pytorch` checkpoint (`tabfm` 1.0.1, 8-member ensemble, frozen; the package exposes no fine-tuning interface, so TabFM has no fine-tuned condition). It is the one deviation from "full training context": its extraction, including its cross-fitted per-fold training posteriors, caps the labelled context at 4,096 stratified training rows, which binds only on larger training splits (the primary population's largest has 4,409 rows). TabPFN and TabICL condition on the full training split throughout.

Version-matched fine-tuning baseline. The fine-tuning interface instantiates the smaller TabPFN v2.5 model (10.7M parameters), while the frozen backbone elsewhere is v3, so we re-extracted a same-version frozen v2.5 baseline (the fine-tuning estimator at zero learning rate) on the audit population. It performs almost identically to frozen v3 natively (46.1 vs. 45.2 test macro-F1) and leaves similar residuals. Against it, standard cross-entropy fine-tuning does not improve native validation macro-F1 (44.6 vs. 45.1; paired $-0.48\,[-1.58, +0.45]$) and leaves the class-selection residual unchanged ($+0.40\,[-1.08, +2.00]$), while class-balanced sampling reduces it by $-4.00\,[-6.32, -1.77]$: the reduction is specific to class-balanced training, not to fine-tuning effort.

Bootstrap and seeds. Intervals resample datasets (10^4 replicates, the dataset as unit); five-seed experiments average seeds within each dataset before the dataset bootstrap. The frozen-model probabilities come from a single inference, so they and every deterministic correction (class-prior shift, prior-ratio, per-class bias, ridge, all DistPFN variants, EM/BBSE, temperature scaling) are seed-invariant; only the MLP-based elements vary with initialization (within-dataset five-seed SD 1.8–3.0 macro-F1), absorbed by the five-seed averages. Table 12 reports the measured fit and prediction-time cost of each correction.

Statistical inference and multiplicity. Bootstrap intervals recomputed in different analysis passes can differ in the final digit; point estimates agree throughout. We control multiplicity separately for the primary decomposition (two components $\times$ three models), the cross-family comparison, and the four fine-tuning contrasts, each Holm-adjusted within its family; mechanism metrics and performance bands are secondary, reported with descriptive intervals and a monotonic-trend test. The DistPFN analysis is a post-hoc external validation, Holm-adjusted within its own family: the validation-tuned class-prior shift exceeds dataset-level DistPFN-T by $+2.11$ to $+3.55$ macro-F1 (label-tuned against label-free), and the ridge residual correction adds significant macro-F1 beyond DistPFN-T on every backbone ($p \le 2 \times 10^{-4}$), with the ranking-AUC increment strongest for TabFM, the only one surviving the most conservative pooled correction ($p = 0.0026$). We reserve confirmatory language for the primary families and treat the rest as robustness or mechanism checks.

Populations, gates, and scoring conventions. Table 10 groups the dataset populations by analysis. Gate "fixed" means the primary seed-42 gate (validation macro-F1 < 0.60 from the uncorrected validation predictions, determined once per model); "own" means the same rule applied to that family's own validation predictions; "recomp." means the run re-derived the < 0.60 gate from its own (uniformly scored) cached validation predictions, so datasets near the threshold can enter or leave. Convention P scores macro-F1 over the classes present in the test labels or predictions (scikit-learn's macro average; the primary decomposition harness); convention A scores over all C classes with absent classes counted as zero (the DistPFN and divergence pipeline, and the tree and FT rows of the matched regressions). The two conventions are why the TabICL class-prior shift reads $+3.52$ in Table 14 and $+3.21$ in Table 13: both conventions select the same τ on all 77 datasets, while TabPFN and TabFM are unaffected. Table 10 summarises one row per population family, with its tables, figures, population sizes and notes.

Table 10: **Population and convention ledger (summary).** One row per population family; n per backbone as TabPFN/TabICL/TabFM unless named; cap = total-row cap; feat. = featurization filter required; det. = deterministic given the cached predictions.

population family	n	gate	cap	feat.	seeds	conv.	analyses
Primary $\leq$8K featurized	76/77/74	fixed	8K	yes	det./5	P	13
Score-only uncapped	84/83/79	fixed	none	no	det.	P/A	2
Own-gate recomputed $\leq$8K	75/76/85 and 78/79/87	recomp.	8K	yes/no	det./5	P	7
$\leq$20K sensitivity	81/80/79	fixed	20K	yes	det.	P	3
Alternate splits 101/202	TabPFN 70/67; TabICL 69/64; TabFM 66/63	re-applied	8K	yes	det.	P	1
Fine-tuning-eligible	71 TabPFN, 76 TabICL	fixed	8K	yes	5	P	3
Cross-fitted probe extraction	126 = 68+58 (72 TabFM layerwise; restoration 50)	fixed	8K	yes	3 splits	P	3
Pretraining-prior small models	72 of TabICL's 77	fixed	8K	yes	2 train seeds	P	1
TabPFN-3.5	fixed = 76, reselected = 73	fixed/re-applied	8K	yes	det.	P	1
Wide-cache bands and spectrum	all cached $C \geq 2$ datasets	banded	8K	mixed	det.	P	2

C.1 Component-level timing

Table 12 measures the correctors on 76 low-performing TabPFN datasets with cached backbone probabilities. On four CPU cores, the class-prior shift and ridge correction together take a median 0.973 ms to score 1,000 test rows in a 21-dataset subsample enriched for wide tables; the corresponding population-wide component measurements appear in the table. Fitting both corrections takes a median 0.062 s once cross-fitted probabilities and indicators are available. The common indicator-building call takes a further median 0.025 s per dataset for training, validation, and test features together. Correction inference therefore measures only the work *after* backbone inference and feature construction.

We also time the full correction on cached scores, now including the feature encoding of new rows (Table 11). Feature construction accounts for most of the median 30.66 ms per 1,000 rows; the two score operations together take less than 1 ms at their individual medians. The complete corrector measurement includes an argmax and uses a separate timing script, so its component medians need not sum to the median total and need not match those in Table 12.

Table 11: **Prediction cost after backbone scores exist, including feature construction.** Milliseconds per 1,000 test rows on 76 low-performing TabPFN v3 datasets, using cached backbone posteriors. The full correction includes featurization, class-prior shift, ridge score adjustment, and argmax; components and total are timed separately over datasets.

operation	mean	median	90th percentile
test-row feature construction	69.38	30.18	160.20
class-prior shift	0.173	0.118	0.372
ridge score adjustment	0.837	0.557	1.244
full correction (including argmax)	70.39	30.66	161.62

For context, a separate backbone timing harness scores 1,000 rows on the same 21 datasets, with 8 ensemble members and training contexts of 77–4,409 rows. TabPFN v3 takes a median 844 ms on an A100 and 19,260 ms on four CPU cores; the other backbones appear in Table 12. The CPU measurements use matched hardware. Ratios to A100 backbone times compare CPU correctors with GPU inference and should be read as scale comparisons rather than complete prediction times. Each timing uses one actual 1,000-row batch, with the training context processed on each scoring call. The harness reproduces GPU cached probabilities to within 7×10^{-7} for TabPFN v3, TabPFN-3.5, and TabICL.

Table 4 complements these cached-score measurements with an A100 measurement of the cross-fitted TabPFN fitting path. The shared context-loading and validation/test-scoring pass takes a median 3.56 s; obtaining out-of-fold training scores and fitting ridge takes an additional 11.65 s with four folds or 6.18 s with two. These measurements include the backbone work excluded by the cached-score fit benchmark above. Because the corrector reads cached posteriors and raw features, its measured cost does not depend directly on the backbone that produced those scores. Comparing its 30.66 ms median against separately measured A100 scoring medians suggests a roughly 2.5–11.4% share of combined scoring-plus-correction time for TabFM through TabICL; these percentages combine different dataset

samples and are estimates rather than jointly timed query latencies. TabFM CPU inference was too slow for the older harness (771 s on its smallest context), and its 10-class checkpoint limit excludes one sampled dataset; Table 12 excludes checkpoint loading.

Table 12: **Measured cost of the three corrections** on the primary TabPFN low-performing population ($n = 76$ datasets), with the frozen backbone's own scoring cost for the same 1000-row batch beneath. *Fit* is per dataset and includes that object's own validation selection; *inference* is the extra work of turning cached scores into corrected predictions, so it starts after the backbone's probabilities exist. Corrector timings are CPU only on 4 pinned cores; *setup* in the backbone block is the one-off cost of constructing the classifier and fitting it on the training context. Medians with interquartile ranges across datasets. Protocol, hardware pairing and limits are given above.

element	extra params (median)	fit s / dataset median [IQR]	inference ms / 1000 rows median [IQR]	overhead (× cached)
cached backbone scoring	—	—	0.080 [0.079, 0.093]	1.00×
class-prior shift	1	0.052 [0.051, 0.053]	0.105 [0.104, 0.135]	1.32×
ridge residual correction	$d \times C$ (≈385)	0.010 [0.008, 0.028]	0.296 [0.174, 0.919]	3.35×
flexible recoverability benchmark	≈ 98,600	0.53 [0.30, 1.49]	1.69 [1.37, 2.45]	19.1×

the frozen backbone's own cost for that same 1000*-row batch* (21 sampled datasets)

backbone	4 CPU cores (matched) setup s / dataset	4 CPU cores (matched) scoring ms / 1000 rows	1 A100 setup s / dataset	1 A100 scoring ms / 1000 rows
TabPFN v3	0.38	19,260	0.56	844
TabICL 2.1.1	0.31	12,155	0.39	238
TabPFN-3.5	not run		1.29	592
TabFM 1.0.1	not measurable		0.07	1,177

Populations differ by analysis; Table 10 gives the gate, row cap, featurization filter, seeds and scoring convention for each.

D Additional comparisons among post-hoc corrections

The two restricted corrections together come within one macro-F1 point of the flexible benchmark on TabPFN, slightly exceed it on TabICL, and leave +1.50 points of its gain unexplained on TabFM. The flexible recoverability benchmark improves test macro-F1 by 7.65, 5.73, and 9.71 points for TabPFN, TabICL, and TabFM on their low-performing datasets (Table 13; all three dataset-bootstrap intervals exclude zero). Table 13 reports all variants, grouped into global class-selection controls (Block A), row-dependent branches from the class-prior shift (Block B), and deployment controls (Block C), with the flexible recoverability benchmark as the reference. Increments in Blocks A and B are measured beyond the one-parameter class-prior shift, the common parent of the row-dependent branches. AUC is computed on the unnormalized per-class log-probability scores (Section 3), where the class-prior shift is exactly zero by construction (Fact 1).

Population and scoring notes. TabICL's AUC column aggregates the 75 datasets with archived AUC-track rows; on the full 77-dataset population the ridge residual correction's AUC gain is $+3.49\,[+0.69, +7.19]$. TabICL's class-prior shift also differs slightly from the value in the DistPFN comparison (Table 14) through the scoring convention alone (Appendix C).

More flexible or class-wide alternatives do not reproduce the row-ordering gain. A nonlinear correction reading only the model's own scores does not improve row ordering and lowers macro-F1 on all three models, a freely fitted per-class bias underperforms the one-parameter class-prior shift, and the row-level weight and dataset-level safeguard add little average macro-F1, serving mainly to prevent deployment regressions. Feature ranges are the large lever.

Single-instance DistPFN variants. These are stronger than their dataset-level counterparts (Table 14), but the row-ordering result is unchanged: After single-instance DistPFN-T,

Table 13: **Full controlled decomposition on the primary low-performing datasets** ($n = 76/77/74$; five seeds averaged within each dataset; higher is better), reported as TabPFN / TabICL / TabFM. "Total" is the cumulative macro-F1 gain over the uncorrected prediction, "Δ shift" the increment beyond the class-prior shift (blank where not a direct child), "AUC" the log-probability one-vs-rest ranking-AUC gain. The centered-linear-feature-range row is the ridge residual correction of the main text.

variant	total macro-F1 gain	Δ shift	log-probability AUC gain
Block A — global class-selection controls			
uncorrected	0.00 / 0.00 / 0.00	—	0.0 / 0.0 / 0.0
class-prior shift	+4.53 / +3.21 / +3.05	—	+0.00 / +0.00 / +0.00
general per-class bias	+2.83 / +0.81 / +1.62	−1.70 / −2.40 / −1.43	—
Block B — row-dependent branches (parent: class-prior shift)			
+ centered linear feature ranges	+6.93 / +6.50 / +8.21	+2.41 / +3.29 / +5.16	+2.10 / +3.58 / +4.30
+ nonlinear feature-range-only	+8.10 / +6.66 / +7.49	+3.57 / +3.45 / +4.44	+1.60 / +2.48 / +4.38
+ nonlinear scores-only	+3.35 / +1.88 / −0.04	−1.17 / −1.33 / −3.09	−0.28 / +1.15 / −0.24
+ nonlinear combined	+7.29 / +5.35 / +7.89	+2.76 / +2.14 / +4.84	+1.28 / +2.01 / +3.87
Block C — deployment controls			
+ row weight	+6.95 / +6.55 / +7.11	—	+1.66 / +2.52 / +4.18
+ dataset-selection gate	+7.23 / +6.25 / +7.81	—	—
flexible recoverability benchmark	+7.65 / +5.73 / +9.71	—	—

Table 14: **Class-wide corrections change class selection but not row ranking; row-specific correction improves both.** Test macro-F1 gains on the low-performing populations ($n = 76/77/74$; 95% dataset-bootstrap intervals). ridge correction = ridge residual correction; ranking-AUC effects are on the per-class score channel, listed TabPFN / TabICL / TabFM, and class-wide rows are float-exactly zero by construction (Fact 1). TabICL's shift differs slightly from Table 13 by scoring convention.

method	TabPFN ΔF1	TabICL ΔF1	TabFM ΔF1	ranking AUC effect
class-wide (cannot reorder rows)				
Dataset-level DistPFN	−1.15 [−2.36, −0.07]	−0.48 [−1.30, +0.42]	−0.57 [−1.51, +0.19]	0 / 0 / 0
Dataset-level DistPFN-T	+0.97 [−0.74, +2.57]	+1.41 [+0.15, +2.88]	+0.89 [−0.42, +2.19]	0 / 0 / 0
class-prior shift	+4.53 [+2.59, +6.51]	+3.52 [+1.94, +5.20]	+3.05 [+1.34, +4.72]	0 / 0 / 0
single-instance DistPFN variants				
Single-instance DistPFN	+2.94 [+1.37, +4.53]	+3.16 [+1.81, +4.62]	+2.67 [+1.36, +4.02]	0 / 0 / 0
Single-instance DistPFN-T	+2.47 [+0.63, +4.34]	+3.60 [+2.02, +5.27]	+2.26 [+0.77, +3.70]	−0.02 / − 0.05 / + 0.03
row-specific (can reorder rows)				
ridge correction beyond dataset-level DistPFN-T	+5.99 [+3.38, +9.02]	+5.32 [+2.26, +9.16]	+6.51 [+3.55, +9.76]	+2.04 / + 3.45 / + 4.29
ridge correction beyond class-prior shift	+2.41 [+0.61, +4.64]	+3.29 [+0.63, +6.57]	+5.16 [+2.37, +8.51]	+2.10 / + 3.58 / + 4.30

the only DistPFN variant here that can itself vary across rows, the ridge residual correction adds +4.17 to +5.46 macro-F1 and +1.88 to +3.67 ranking-AUC points. The component therefore survives both class-wide and row-varying posterior adjustment.

Head-to-head inference. Each corrected frozen model leads corrected XGBoost and CatBoost by +0.91 to +2.75 macro-F1 on average, but after Holm correction across the six comparisons only TabICL remains significant against both trees.[1] The component comparison is stronger: all six frozen-versus-tree class-prior shift differences remain Holm-significant ($+4.53/+3.21/+3.05$ frozen against $+0.05/-0.43/+0.02$ and $+0.48/+0.45/+0.46$ for the trees).

Tuned trees. The absolute comparison above uses library-default trees. Repeating the native, shift and row-correction arms on the 20-trial tuned XGBoost and CatBoost caches (single seed; 95 datasets each, of which 65 and 70 are low-performing for the tuned tree) gives native 47.7/48.3 macro-F1, shift gains of $-0.64\,[-2.17, +0.75]$ and $+0.22\,[-0.75, +1.10]$, and row-correction increments of $+0.88\,[+0.10, +1.95]$ and $+2.67\,[+0.66, +5.56]$; neither shift gain approaches the frozen range of +3.05 to +4.53. On the datasets shared with each frozen

[1] A broader re-analysis applying the identical two-stage correction to both sides on every shared low-performing dataset (231 family–dataset pairs, 94 datasets) reproduces this per tree ($+1.76\,[-0.10, +3.63]$ against XGBoost, $+0.75\,[-1.15, +2.63]$ against CatBoost), but the advantage does not survive an oracle best-of-two tree chosen per dataset ($-1.47\,[-3.34, +0.31]$). There is no small-data advantage either.

Table 15: **Absolute test macro-F1 (×100) against directly trained baselines, with the same corrections applied to the trees.** Each row uses the frozen backbone's own low-performing population; tree results use the same datasets and averaging protocol, and the "corr." columns apply both corrections to the trees, selected on validation exactly as for the frozen backbones. Bold marks a Holm-significant advantage (paired Wilcoxon, six tests) over both corrected trees; the parenthesised flexible benchmark is a reference for recoverable performance. **Rows use different dataset sets and are not comparable.**

		frozen backbone			XGBoost		CatBoost		
backbone	n	uncorrected	+class-prior shift	+class-prior shift+ridge correction	native	corr.	native	corr.	*flexible benchmark*
TabPFN	76	46.26	50.78	53.19	49.81	51.14	49.62	51.58	(53.90)
TabICL	77	48.24	51.45	**54.74**	51.24	51.99	49.81	52.44	(53.96)
TabFM	74	46.36	49.41	54.57	51.64	52.96	51.61	53.66	(56.06)

model's primary population the corrected frozen models edge or tie the corrected tuned trees, while natively the tuned trees lead by 2.1–6.1 points, so tuning does not change the picture of Table 15. These single-seed gains differ from the five-seed tuned gains of Appendix F.2, which hold the trees' primary populations fixed. A search for posterior-only cross-dataset gating rules that would avoid the datasets each correction harms found none that improved on ordinary validation selection.

We read the decomposition as identifying *which information channel* repairs each component, not as an additive partition of a fixed ceiling: the split is order-dependent, since fitting the output-level feature head of Section 4.4 first and re-selecting τ shrinks the class-selection gain, though not significantly. Neither correction is uniformly beneficial either, the class-prior shift lowering test macro-F1 on 21–30% of datasets and the row correction on 20–22%.

E Minority-class probability and decisions

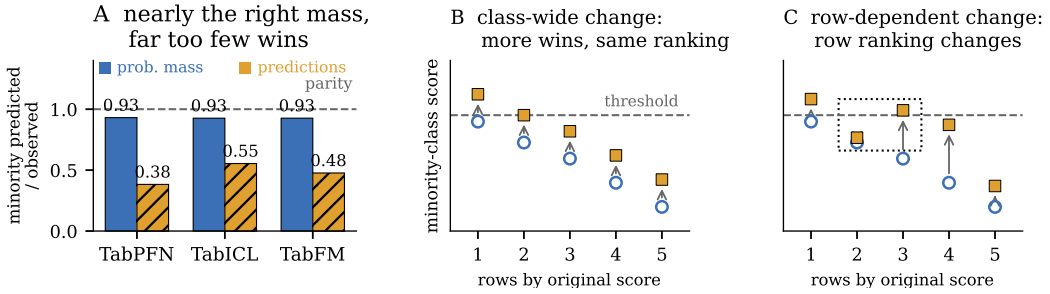

Figure 3: **What class-wide and row-dependent corrections change.** *(A)* On low-performing datasets minority classes receive about the expected probability mass (pooled 0.93 per backbone, wide per-dataset spread; Appendix E) but win the argmax much less often than they occur. *(B)* A class-wide change moves every row of a class equally, changing how often that class wins without changing row ranking. *(C)* A row-dependent change moves rows differently and can change their ranking.

Mass is approximately right (pooled 0.93, mean 0.99) while selection is not (pred/obs 0.37–0.53, across-dataset means). After the shift, pred/obs reaches parity and minority *precision* rises. Table 16 reports per-model minority evidence behind the class-selection component, before → after the class-prior shift, each on that model's own low-performing datasets ("pred/obs" is minority predicted-over-observed frequency). The mass ratio is the average, over the minority classes c present in the test labels, of $\big(\sum_x p_c(x)\big)/\#\{x : y(x) = c\}$, averaged again across datasets. Its per-dataset dispersion is substantial: the mean is 0.99 on every backbone, with an interval including exact parity, but the median dataset sits at 0.93–0.95, the interquartile range is about $[0.84, 1.07]$, and only 36–46% of datasets fall within ±10% of parity. The hard-prediction discrepancy is much larger: before correction all three models give minority classes approximately the correct aggregate mass (mean mass ratio 0.99) yet select them well below their observed rate (pred/obs 0.37–0.53); both are across-dataset means of per-dataset ratios, and the pooled values quoted in Section 3 are

0.93 and 0.38–0.55 on the same datasets. The class-prior shift moves the prediction ratio to near parity and roughly doubles minority recall while minority *precision* also rises, so the gain is better minority selection rather than indiscriminate minority prediction.

Table 16: Per-model minority-class metrics on each model's own low-performing datasets, before → after the class-prior shift (seed-42 cached predictions; brackets are 95% intervals on the post-correction value or the change). "mass" is the aggregate predicted-to-observed minority probability-mass ratio; the final column shows the shift lowering accuracy and raising log-loss and Brier.

model	n	mass	pred/obs	precision	recall	F1	Δacc / ΔLL / ΔBrier
TabPFN	84	0.99	0.37→1.22	34.1→39.4	20.1→41.2 [35.9, 46.6]	22.5→37.0	−2.18 / +0.058 / +0.037
TabICL	83	0.99	0.53→1.18	36.8→41.2	26.5→42.2 [36.9, 47.4]	28.8→39.0	−1.92 / +0.056 / +0.034
TabFM	79	0.99	0.51→1.14	34.1→39.5	23.4→39.8 [34.9, 44.8]	25.6→36.6	−1.76 / +0.051 / +0.034

Read as probabilities, the row-corrected channel is close to forecast-neutral, unlike the shift-only channel. Against the native forecast its accuracy and Brier score do not separate, and the only significant residual is TabICL log-loss +0.038 [+0.001, +0.080]. Keeping the native probabilities as the forecast remains the safe default.

"Approximately the right total probability" is a population-mean statement, not a per-dataset guarantee; optimized for macro-F1, the shift does not improve accuracy or proper scoring, with accuracy falling (−1.8 to −2.2) and log-loss and Brier score rising on every model.

E.1 Probability mass on losing rows

The frozen models hold roughly the right mass but leave 69–80% of it on rows the class does not win, against 48–53% for the trees—we call this *hedging*. Their minority evidence is not detectably worse (prec@k 0.42–0.47 against 0.43–0.45). We measure *where* the minority probability of Section 3 sits, for every family with cached scores, on each family's own low-performing datasets (n in Table 17). For each minority class c present in the test labels (count at or below the median class count) we compute, from the native test probabilities, the mass ratio $\sum_x p_c(x)/\#\{x : y(x) = c\}$, the *losing share* $\sum_{x:\arg\max \neq c} p_c(x)/\sum_x p_c(x)$ of the class's probability that sits on rows it does not win, and the precision at $k = \#\{x : y(x) = c\}$ of the rows ranked by $p_c(x)$, a ranking quantity no class-wide shift can change. Each is averaged over a dataset's minority classes and then across datasets (dataset bootstrap, 10^4 replicates).

Table 17: **Where minority probability sits, by family** (each family's own low-performing datasets under the 8K cap without the featurization filter, single seed-42 pipeline, so populations and tree gains differ from Table 21's five-seed featurized ones; across-dataset means with 95% dataset-bootstrap intervals). pred/obs is the minority predicted-over-observed frequency; the losing share is the fraction of minority probability on rows the class does not win; prec@k is the precision of the k top-ranked rows, k the class's observed count; the last column is the class-prior shift gain on the same datasets.

family	training	n	mass ratio	pred/obs	losing share	prec@k	shift gain
TabPFN	pretrained, frozen	78	0.99 [0.93, 1.06]	0.38 [0.29, 0.47]	0.80 [0.75, 0.85]	0.42 [0.37, 0.47]	+4.67 [+2.77, +6.54]
TabICL	pretrained, frozen	79	0.99 [0.92, 1.06]	0.54 [0.45, 0.64]	0.69 [0.63, 0.75]	0.47 [0.42, 0.53]	+3.35 [+1.70, +5.06]
TabFM	pretrained, frozen	85	0.98 [0.91, 1.05]	0.53 [0.44, 0.63]	0.70 [0.64, 0.75]	0.44 [0.39, 0.49]	+2.84 [+0.80, +5.00]
logistic regression	task-trained	118	1.06 [0.97, 1.18]	0.62 [0.54, 0.71]	0.65 [0.60, 0.70]	0.42 [0.38, 0.46]	+3.25 [+1.88, +4.63]
LDA (empirical prior)	task-trained	130	1.21 [1.02, 1.49]	0.98 [0.66, 1.49]	0.59 [0.54, 0.64]	0.42 [0.38, 0.45]	+2.63 [+1.58, +3.71]
FT-Transformer	from scratch	63	1.33 [1.05, 1.68]	0.54 [0.42, 0.66]	0.75 [0.69, 0.81]	0.36 [0.31, 0.41]	+1.17 [+0.22, +2.18]
CatBoost	task-trained	77	0.89 [0.83, 0.94]	0.68 [0.60, 0.76]	0.53 [0.47, 0.60]	0.45 [0.39, 0.51]	+1.10 [+0.27, +1.92]
XGBoost	task-trained	78	0.87 [0.80, 0.94]	0.73 [0.64, 0.81]	0.48 [0.42, 0.54]	0.43 [0.38, 0.48]	−0.15 [−1.18, +0.80]

Hedging versus committing. The trees allocate less than the observed mass to minority classes (0.87–0.89) and select them at a rate close to that mass (0.68–0.73), so their decisions track their probabilities. The in-context models allocate the right mass (0.98–0.99) but select minorities at 0.38–0.54 of their rate (across-dataset means), with 69–80% of that mass on rows the class does not win against 48–53% for the trees, and their minority *evidence* is not worse (prec@k 0.42–0.47 against 0.43–0.45). On the 84 datasets low-performing for at least one frozen model and at least one tree (values averaged within each group), the frozen-minus-tree differences are +0.11 [+0.07, +0.15] in mass ratio, −0.19 [−0.25, −0.13] in

pred/obs, $+0.21\,[+0.17, +0.26]$ in losing share, and $+0.02\,[-0.00, +0.05]$ in prec@k, with a shift-gain difference of $+2.85\,[+1.49, +4.24]$. Logistic regression hedges like the in-context models and gains +3.25; LDA over-allocates mass and already selects at parity, yet 59% of its mass is on losing rows and it still gains +2.63. The contrast is therefore between soft-posterior models and committed tree ensembles, and that learner families differ systematically in posterior shape is classical (Niculescu-Mizil and Caruana, 2005).

Hedging statistically absorbs the cross-family difference. Within every soft-posterior family the shift gain rises with the losing share and the trees' correlations are much weaker; within dataset, the frozen-minus-tree difference in shift gain tracks the difference in losing share with an intercept 4% of the raw +2.85 gap ($n = 84$); pooled over all eight families, the shift pays when mass is hedged *and* the evidence is good. Restricting to the frozen models and the trees, controlling for the two statistics moves the gap as Table 18 shows. The FT-Transformer shows that the statistic is not a restatement of the gain: it hedges as much as TabPFN (0.75) but its minority evidence is poor (prec@k 0.36) and its gain is small (+1.17; its validation data also selected its architecture and early stopping, Appendix E.3). What the statistic adds over pred/obs is a split rather than new evidence: one minus the losing share is approximately argmax recall when the mass ratio is near one (0.80/0.69/0.70 here against $1 -$ recall $= 0.799/0.735/0.766$ from Table 16, on different populations), an approximation that fails where the mass ratio departs from one (FT-Transformer 1.33, trees 0.87–0.89). Appendices E.1.1 and E.1.2 bound two candidate explanations and test which property of a hedged posterior carries the gain.

Table 18: **The frozen-minus-tree gap in class-prior shift gain, with and without hedging controls** ($n = 397$ family–datasets over 100 datasets; dataset-clustered 95% intervals). Each specification is a same-sample description: the controls and the gain come from one test split.

specification	frozen − tree gap in shift gain
no control	$+3.13\,[+1.82, +4.51]$
+ losing share	$+1.22\,[-0.20, +2.53]$
+ losing share and prec@k	$-0.06\,[-1.75, +1.45]$

The losing share is not independent evidence of under-selection, since it approximately restates argmax recall, and every regression here uses the same test-split statistics that yield the shift gain, so all of it describes an association rather than establishing a cause.

E.1.1 Ensembling and context-size controls

Inference-time averaging raises the losing share by only $+0.009\,[+0.003, +0.015]$, about 4% of the frozen-minus-tree gap, and halving the labelled context does not raise the class-selection gain relative to refit trees ($-0.69\,[-1.34, -0.04]$ per halving). Against a single member on the same cross-fitted rows (68 TabPFN and 58 TabICL datasets), averaging raises the losing share by only $+0.009\,[+0.003, +0.015]$ and leaves the shift gain unchanged ($+0.13\,[-0.50, +0.77]$); a single member already hedges. In a pre-registered context-size intervention (stratified subsamples of 1, 0.5 and 0.25 of the training split, fresh inference, validation and test fixed; 63 TabPFN and 59 TabICL datasets with at least 200 training rows; trees refit on the same rows; dataset-clustered bootstrap), halving the labelled context does not raise the class-selection gain in TabPFN and TabICL relative to the refit trees. Per halving, given native test macro-F1, the frozen shift gain changes by −0.94 against −0.25 for the trees (difference $-0.69\,[-1.34, -0.04]$), and the frozen losing share is flat ($-0.001\,[-0.015, +0.012]$) while the trees' falls ($-0.022\,[-0.036, -0.006]$; difference $+0.021\,[+0.003, +0.037]$).

Both arms bound two specific mechanisms on two backbones; they do not identify what does produce the hedging.

E.1.2 Concentrating probability mass near the winning class

Hedged mass just below the argmax carries it: at matched entropy the contrast is $\Delta(1) = +1.56\,[+1.11, +2.05]$, or $0.56\,G(0)$ against $G(0) = +2.79\,[+1.93, +3.71]$. At $\alpha = 2$, $\Delta = +0.70\,[+0.16, +1.35]$ meets the pre-registered bar of $0.25\,G(0)$ at the point estimate only

(0.252 [0.07, 0.43]), and not when TabPFN is dropped (+0.49 [−0.05, +1.09], 0.21 $G(0)$; post hoc); at $\alpha = 4$ the entropy match fails and the sign reverses ($\Delta = -0.72$ [−1.22, −0.32]), its temperature sitting at the 0.01 floor on 18 of 227 frozen units against 2 and 5 at $\alpha = 1$ and $\alpha = 2$. We transform each cached posterior in two argmax-preserving ways and rerun the self-consistency null of Appendix E.3 unchanged (labels sampled from the transformed posterior, τ re-selected on synthetic validation, 50 draws with the same seeds in every arm, Table 21's populations). Winner commitment $W(\alpha)$ adds α to the winning logit; entropy-matched temperature $T(\alpha)$ divides the logits by a per-dataset temperature in $[0.01, 1]$, set by bisection so that mean normalized entropy over validation and test rows equals that of $W(\alpha)$. The claim-bearing quantity, fixed before running, is $\Delta(\alpha) = G_T(\alpha) - G_W(\alpha)$ for the pooled in-context models (equal-weight backbone means, dataset-clustered bootstrap with 4000 replicates, 95 datasets; Figure 4). This is consistent with F-measure threshold theory (Lipton et al., 2014; Narasimhan et al., 2014), which places the recoverable decision gain in minority mass just below the argmax, removed faster by winner commitment than by entropy-matched temperature.

The decline of the gain with α is guaranteed by construction, not evidence, since argmax becomes optimal under its own labels as either transform makes the posterior one-hot; the check is weaker at $\alpha = 2$, reversed where the entropy match fails, and silent about where the hedging comes from.

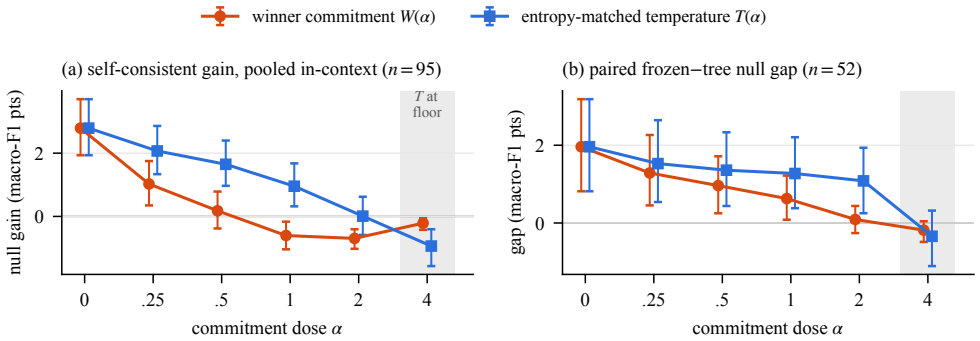

Figure 4: **Commitment dose-response on self-consistent posteriors.** (a) Self-consistent class-prior shift gain for the pooled in-context models (95 datasets) under winner commitment $W(\alpha)$ and entropy-matched temperature $T(\alpha)$; both preserve every row's argmax, so both gains must vanish as the posterior becomes one-hot and only their gap at matched entropy is informative. (b) Paired frozen-minus-tree null gap on the 52 common datasets. Macro-F1 points with 95% dataset bootstrap intervals; the α axis is not to scale. Shaded: $\alpha = 4$, where the entropy match fails.

E.2 Validation tuning and self-selected shifts

The gain also appears when τ is chosen from the model's own validation posterior; this choice matches or exceeds the tuned shift on all three backbones, and the parameter-free $\tau = -1$ member recovers only 0.57/0.54/0.74 of the tuned gain. On the in-context models, adjustments of increasing flexibility recover diminishing amounts. The parameter-free balanced-softmax shift ($a = -\log \pi$; Ren et al., 2020) is the $\tau = -1$ member of the validation-tuned class-prior shift family ($a = \tau \log \pi$), and TabPFN's `balance_probabilities` option computes exactly this arm, so Table 19 reports what that flag would recover on our populations (applying $\tau = -1$ to the cached scores rather than re-running the flag). Always applied, it recovers roughly half to three quarters of the tuned gain (0.57/0.54/0.74 on TabPFN/TabICL/TabFM), and 0.77/0.74/0.76 when applied only if it improves validation macro-F1. The paired tuned-minus-τ=−1 difference excludes zero on TabPFN alone, and only 27–31% of datasets select a τ within 0.25 of −1 (median tuned τ −0.80/ − 0.70/ − 0.70), so the free parameter is not idle. On the trees $\tau = -1$ does not help (XGBoost −0.33; median tuned τ 0.00). The general per-class bias ($a = b \in \mathbb{R}^C$)—despite C free parameters—does not improve on the tuned shift, consistent with overfitting the small validation sets. This is the same qualitative pattern as the fixed-population decomposition (Section 3), measured under the family-comparison protocol.

Table 19: **The parameter-free $\tau = -1$ member against the validation-tuned shift.** Test macro-F1 gains over the native model on the score-only populations (validation macro-F1 < 0.60, $\leq$ 8K rows, $C \geq 2$, no featurization filter; $n = 78/79/87/78/78$), with 95% dataset-bootstrap intervals (10^4 replicates). "val-gated" applies $\tau = -1$ only where it improves validation macro-F1; "tuned $-$ (τ=-1)" is the paired difference; the last column is the share of datasets whose tuned τ lies within 0.25 of -1. TabPFN's row is what its `balance_probabilities` flag would recover. This population is the score-only gate of the featurization-free light run, slightly larger than the seed-42 pipeline of Table 17 and reconciled with it in Table 10.

family	n	tuned τ	$\tau = -1$	$\tau = -1$, val-gated	tuned $-$ (τ=-1)	$\lvert\tau^* + 1\rvert \leq 0.25$
TabPFN	78	$+4.67\,[+2.77, +6.54]$	$+2.68\,[+0.57, +4.72]$	$+3.58\,[+2.04, +5.18]$	$+2.00\,[+0.66, +3.36]$	31%
TabICL	79	$+3.35\,[+1.70, +5.06]$	$+1.82\,[-0.13, +3.66]$	$+2.47\,[+1.42, +3.61]$	$+1.53\,[-0.01, +3.23]$	27%
TabFM	87	$+2.78\,[+0.80, +4.90]$	$+2.06\,[+0.17, +4.12]$	$+2.12\,[+0.75, +3.45]$	$+0.72\,[-0.49, +1.85]$	31%
XGBoost	78	$-0.15\,[-1.18, +0.80]$	$-0.33\,[-1.45, +0.68]$	$+0.35\,[-0.42, +1.05]$	$+0.18\,[-0.97, +1.35]$	9%
CatBoost	78	$+1.08\,[+0.26, +1.91]$	$+0.64\,[-0.72, +2.34]$	$+0.72\,[+0.12, +1.37]$	$+0.44\,[-1.11, +1.75]$	18%

Selecting τ without validation labels. The tuned shift fits τ to the validation labels. We also select τ from the model's own validation posterior: draw 50 label vectors from it, tune τ for macro-F1 on each over the same 41-point grid, and apply the median. This sampled plug-in approximates an expected-macro-F1 decision (Dembczyński et al., 2011; Ye et al., 2012); it still uses training labels for π and validation labels to select the dataset population. As a variance control, we resample the observed validation labels the same number of times. On the primary populations posterior selection matches or exceeds the tuned shift on all three backbones (Table 20): the paired posterior selection-minus-tuned interval covers zero with a non-negative point on each, and posterior selection exceeds the $\tau = -1$ member by $+3.13\,[+1.71, +4.74]$, $+2.66\,[+0.96, +4.54]$ and $+1.87\,[+0.60, +3.26]$. The score-only populations replicate this (posterior selection minus tuned $+0.97\,[-0.38, +2.41]$, $+1.07\,[-0.26, +2.46]$, $+0.17\,[-1.74, +1.77]$). On the trees the rule does not help. A label-bootstrap control narrows the gap to the validation-tuned shift, but neither reduced tuning variance nor additional information in the model posterior explains the apparent advantage consistently across all three backbones.

Table 20: **Selecting τ without validation labels.** Test macro-F1 gains over the native model on the primary populations (in-context models: the fixed low-performing gate of Table 13; trees: their own gate and datasets of Table 24), with 95% dataset-bootstrap intervals (10^4 replicates). "tuned": τ tuned on the real validation labels; "posterior: median of 50 values of τ, each tuned on labels sampled from the model's validation posterior; "bootstrap: the same on resamples of observed validation labels. The last two columns are paired differences. In every column π uses training labels and the gate uses validation macro-F1; tree rows use the single seed-42 caches.

family	n	tuned τ	posterior	bootstrap	posterior $-$ tuned	posterior $-$ bootstrap
TabPFN	76	$+4.53\,[+2.60, +6.50]$	$+5.63\,[+3.87, +7.43]$	$+5.12\,[+3.14, +7.15]$	$+1.10\,[-0.31, +2.58]$	$+0.50\,[-0.81, +1.82]$
TabICL	77	$+3.21\,[+1.53, +4.95]$	$+4.32\,[+2.71, +6.11]$	$+3.12\,[+1.67, +4.64]$	$+1.10\,[-0.26, +2.54]$	$+1.20\,[+0.15, +2.35]$
TabFM	74	$+3.05\,[+1.36, +4.70]$	$+3.99\,[+2.57, +5.43]$	$+3.49\,[+2.02, +4.97]$	$+0.94\,[-0.34, +2.26]$	$+0.50\,[-0.55, +1.58]$
XGBoost	78	$-0.15\,[-1.18, +0.80]$	$-0.24\,[-0.93, +0.44]$	$-0.37\,[-1.52, +0.67]$	$-0.08\,[-1.29, +1.12]$	$+0.14\,[-0.99, +1.25]$
CatBoost	77	$+1.10\,[+0.27, +1.92]$	$+0.38\,[-0.24, +1.00]$	$+1.49\,[+0.61, +2.31]$	$-0.71\,[-1.60, +0.11]$	$-1.11\,[-1.85, -0.40]$

A wider plug-in panel on the full uncapped populations ($n = 84/83/79$) confirms the picture. The class-prior shift ($+4.86/+3.54/+3.33$ macro-F1) is the only method whose interval excludes zero on all three backbones: an L_2-regularized per-class bias is not significant; temperature scaling cannot change any argmax decision and gives exactly $+0.00$ by construction; vector and Dirichlet scaling are null to negative as *plug-in competitors* fit on validation, and Appendix E.5 instead applies them *before* the shift, which does not remove the component; and the EM (Saerens et al., 2002), MLLS (Alexandari et al., 2020) and BBSE (Lipton et al., 2018) label-shift estimators *hurt* by 5–11 macro-F1 points (by 6–12 on the primary decomposition populations), consistent with there being no label shift to estimate.

Divergence control. If the class-selection component were hidden label shift, the class-prior shift gain should track the Jensen–Shannon (JS) divergence between training and test class proportions; it does not. Pooled across backbones ($n = 229$ pairs from 97 canonical datasets, intervals clustered by canonical dataset), the gain is full-size in the *lowest*-divergence quartile (median JS $\approx 4 \times 10^{-4}$ nats): $+6.11\,[+3.64, +8.69]$ macro-F1, versus $+2.93\,[+1.61, +4.28]$ in the upper three quartiles. With controls for model strength,

imbalance, class count, size, suite and backbone, the JS-divergence coefficient is -0.42 F1 per SD ($[-1.19, +0.35]$), so the gain is not positively driven by prior shift. The quartile contrast is confounded with class count, so we do not claim the gain is concentrated at low divergence, only that it is fully present there.

The apparent advantage of posterior selection remains uncertain because the gate selects datasets on the validation macro-F1 that the tuned rule fits, exposing that rule to selection noise avoided by the posterior-sampled alternative, which alone could produce its larger point estimate. We therefore read the result only as a match: on the in-context models, choosing τ does not require validation labels.

E.3 Class decisions under model-sampled labels

81% [58%, 122%], 70% [40%, 138%] and 79% [50%, 153%] of the observed gain for TabPFN, TabICL and TabFM. For each model–dataset pair we sample synthetic validation and test labels from the model's own predicted probabilities, re-select τ on the synthetic validation set, and measure its macro-F1 gain on the independent synthetic test set. The model is correct by construction there, so the gain measures the class-selection component that the model's probability distribution and argmax rule alone can produce (50 simulations per dataset on the canonical 41-point τ grid, deterministic for the frozen models given the fixed split and averaged over five training-seed caches for the directly trained baselines; independent simulation streams move the frozen-family means by at most 0.16 macro-F1 points). Table 21 reports the result; for the directly trained families, observed-minus-null excesses are compatible with zero.

Table 21: **Self-consistency experiment across model families.** "observed" is the class-prior shift gain on the original labels, "null" the gain under labels simulated from the family's own predicted probabilities, "excess" their difference. Intervals are dataset bootstraps (10^4 replicates), with two-sided bootstrap p on the excess.

family (own population)	n	observed	null	excess (obs−null)	pred/obs	max-p	ent/ $\log C$
TabPFN (frozen)	76	+4.53 [+2.59, +6.51]	+3.69 [+2.44, +5.06]	+0.84 [−0.67, +2.38], $p = 0.28$	0.37→0.39	0.647	0.675
TabICL (frozen)	77	+3.21 [+1.53, +4.97]	+2.26 [+1.28, +3.30]	+0.95 [−0.68, +2.59], $p = 0.26$	0.53→0.55	0.670	0.657
TabFM (frozen)	74	+3.05 [+1.34, +4.72]	+2.42 [+1.48, +3.43]	+0.63 [−0.84, +2.06], $p = 0.38$	0.51→0.51	0.656	0.678
TP-BERTa (pretrained, fine-tuned)	136	+4.59 [+2.88, +6.59]	+3.25 [+2.40, +4.24]	+1.33 [−0.48, +3.25], $p = 0.14$	—	—	—
XGBoost	78	−0.32 [−1.22, +0.49]	+0.28 [−0.24, +0.85]	−0.60 [−1.55, +0.27], $p = 0.18$	0.72→0.80	0.773	0.482
CatBoost	77	+0.41 [−0.23, +1.05]	+0.82 [+0.13, +1.58]	−0.41 [−1.30, +0.39], $p = 0.34$	0.68→0.76	0.743	0.513
FT-Transformer (scratch)	61	+0.49 [−0.08, +1.09]	+0.73 [+0.19, +1.40]	−0.25 [−0.91, +0.40], $p = 0.44$	0.59→0.60	0.604	0.715
SAINT (scratch)	104	+0.86 [+0.05, +1.69]	+0.85 [+0.27, +1.51]	+0.01 [−0.67, +0.67], $p = 0.98$	—	—	—

Additional notes. "pred/obs" is the minority predicted-over-observed frequency, observed → null (frozen values on the uncapped populations, as means of per-dataset ratios; the pooled values of Section 3 are 0.38–0.55); "max-p" and "ent/ $\log C$" are mean maximum predicted probability and normalized entropy, not recomputed for TP-BERTa or SAINT (—), which are evaluated on their self-consistency-eligible populations ($n = 136$ and 104), so TP-BERTa's +4.59 observed gain sits on a larger population than the +4.42 of Table 24.

On the 52 datasets common to the frozen models and both trees, the observed frozen-minus-tree class-prior shift difference is +3.05 [+1.77, +4.37], against +1.96 [+0.83, +3.15] under synthetic labels, with residual +1.09 [−0.18, +2.32]: roughly two-thirds of the cross-family difference reappears when each family's probabilities are correct by construction. The same holds on the 45 datasets common to all six families and in both scoring conventions.

Overall confidence does not account for the gain. Frozen models have lower maximum probabilities and higher normalized entropy than the trees, but entropy predicts the synthetic gain poorly across datasets (pooled Spearman $r \approx 0.03$, inconsistent within-family signs), and the FT-Transformer is very diffuse yet has a small component. What tracks the gain is not scalar confidence but how much minority mass sits on rows that lose the argmax, a same-sample description that restates under-selection rather than explaining it (Appendix E.1).

Construction checks. Replacing macro-F1 with accuracy gives the required negative control: argmax maximizes expected accuracy under $y \sim p(x)$ and $\tau = 0$ is available, so no positive population gain is possible, and the synthetic accuracy gains are negative for every

family (-0.5 to -0.9). Temperature scaling gives the complementary probability-quality control: nearly neutral on the real posteriors but harmful on the synthetic, already-correct ones ($+0.03$ to $+0.08$ log-loss, $p = 10^{-4}$). A decision-rule correction surviving the null while a probability calibrator is destroyed by it supports the intended interpretation. The null measures what the decision rule can contribute when the posterior is correct; it does not diminish the practical gains, which are measured against real labels, and it cannot touch the row-ordering results, since an argmax objective mismatch leaves the per-class score order unchanged and so explains none of the ranking-AUC gains ($+2.10/+3.49/+4.30$, TabICL at its full-population value) or the increments beyond DistPFN-T. Nothing here rests on the FT-Transformer alone, which used validation data for architecture selection and early stopping.

The ratio intervals are wide and only TabPFN's lower bound exceeds 50%, and they are statements about the magnitude of the synthetic effect, *not* attribution fractions for real-data error.

E.4 Sensitivity of the self-consistency test

The first detectable prior-direction dose is $s = 0.25$, 0.5 and 1.0, at paired excesses of $+2.22$, $+3.11$ and $+2.86$ macro-F1; the real excesses lie below each backbone's detectable range. The null's resolution is relative to the correction family it searches over. We characterize that limit by injecting class-wide distortions of known magnitude into the cached scores and asking at what size the excess of observed gain over null becomes detectable; labels are never resampled in the observed arm, the real validation and test labels being used throughout while the injection touches only the score matrix. *Prior-direction distortions.* Adding $s \log \pi$ to the logits produces an excess rising monotonically with s on all three backbones. Under a paired dataset bootstrap against the uninjected model, Holm-corrected across doses, the first detectable dose is $s = 0.25$, 0.5 and 1.0 for TabPFN, TabICL and TabFM, at paired excesses of $+2.22$, $+3.11$ and $+2.86$ macro-F1, and the observed excesses on the real models ($+0.84$, $+0.95$, $+0.63$) lie below their own backbone's detectable range.

Orthogonal distortions. A class-wide distortion orthogonal to $\log \pi$ exists only when $C \geq 3$, since for binary tasks the centered class space is spanned by $\log \pi$. On the $C \geq 3$ datasets ($n = 101$ pooled) such distortions produce no detectable excess at any dose while damaging native macro-F1 by 1.58 points *more* on average than the prior-direction injection; the paired gap in excess is $-4.02\,[-6.57, -1.44]$ ($p = 0.003$).

These are sensitivity statements about the test, not a bound on miscalibration and not a null result. The class-wide channel cannot detect class-conditional error orthogonal to the training prior, so a small excess should be read with that limitation in mind.

E.5 Class selection after recalibration

Under the validation-selected calibrator the shift still recovers $+4.52$, $+2.38$ and $+2.73$ macro-F1, and the paired change from the uncalibrated gain is indistinguishable from zero on every backbone. Section 4.2 attributes a substantial part of the class-selection component to the decision rule rather than to probability error. The sharpest objection is that the posteriors are miscalibrated in a way a calibrator richer than temperature scaling would repair. Appendix E.2 evaluates vector and Dirichlet scaling as *plug-in competitors* fit on validation; here we give recalibration its strongest fair form, applying it *before* the class-prior shift and asking whether the component is still there afterwards.

Protocol. For each model–dataset pair we fit four calibrators on the out-of-fold training posteriors by L-BFGS: identity; temperature scaling; vector scaling $z' = a \odot z + b$ with an L_2 penalty on $(a - 1, b)$; and Dirichlet calibration (Kull et al., 2019), a full $C \times C$ matrix scaling of the log-probabilities with ODIR regularization. Penalty strength and the calibrator family are both selected by *validation log-loss*; test is touched once, after selection. We then rerun the class-prior shift and the self-consistency null of Appendix E.3 on the recalibrated posterior, re-tuning τ on the recalibrated validation scores and re-drawing the null labels from the recalibrated distribution. Fitting on the training posteriors does not leak: they

are out of fold, with train macro-F1 below validation and test. The arms are VS (vector scaling), DIR (Dirichlet forced) and SEL (validation-selected).

Table 22: class-prior shift gain and self-consistency null after recalibration ($n = 76/77/74$; test macro-F1; 95% dataset-bootstrap intervals). *native* is the uncalibrated result of Section 4.2; Δobs is the paired change relative to it.

backbone	arm	observed gain	null	null/obs	Δobs vs. native
TabPFN	native	$+4.53\,[+2.61, +6.49]$	$+3.69$	0.81	—
	VS	$+4.46\,[+2.62, +6.41]$	$+3.67$	0.82	$-0.06\,[-1.37, +1.19]$
	DIR	$+4.45\,[+2.53, +6.39]$	$+3.60$	0.81	$-0.08\,[-1.75, +1.63]$
	SEL	$+4.52\,[+2.61, +6.55]$	$+3.58$	0.79	$-0.00\,[-1.37, +1.33]$
TabICL	native	$+3.21\,[+1.50, +4.92]$	$+2.26$	0.70	—
	VS	$+3.53\,[+1.81, +5.28]$	$+3.03$	0.86	$+0.31\,[-0.89, +1.51]$
	DIR	$+4.62\,[+2.04, +7.16]$	$+3.51$	0.76	$+1.41\,[-0.45, +3.58]$
	SEL	$+2.38\,[+0.33, +4.36]$	$+2.58$	1.08	$-0.83\,[-2.13, +0.36]$
TabFM	native	$+3.05\,[+1.36, +4.73]$	$+2.42$	0.79	—
	VS	$+3.45\,[+1.85, +5.07]$	$+3.24$	0.94	$+0.40\,[-0.88, +1.80]$
	DIR	$+4.15\,[+2.53, +5.82]$	$+3.68$	0.89	$+1.10\,[-0.70, +2.95]$
	SEL	$+2.73\,[+1.09, +4.37]$	$+2.83$	1.04	$-0.32\,[-1.65, +1.13]$

The component survives recalibration. The class-prior shift intervals exclude zero on all three backbones under the validation-selected calibrator, narrowly so on TabICL (Table 22). Under vector scaling, one fixed family and so free of family-selection effects, the gain is if anything larger on two backbones. The null share does not move away from the decision rule (79%, 108% and 104% against 81%, 70% and 79% natively), with the excess of observed gain over null statistically zero in every arm.

Forced Dirichlet calibration is the informative comparison. A calibrator selected by log-loss need not change any prediction, and here it largely does not: the selected map leaves every test prediction unchanged on 53/76, 53/77 and 49/74 datasets. Forced Dirichlet calibration, applied regardless of validation log-loss, changes test predictions on 62–69% of datasets and lowers native macro-F1, yet still leaves $+4.45$, $+4.62$ and $+4.15$ points. Recalibration also does what the objection asks on the measure it names: class-wise expected calibration error falls significantly on all three backbones under vector scaling ($p = 0.031$, 0.010, 0.011; Holm-adjusted 0.031, 0.030, 0.030), while log-loss and Brier score improve on none. The class-wise reliability was genuinely improvable, it was improved, and the component is still there — though that calibration improvement holds for the cross-fitted fit only, a half-validation refit at these sizes being dominated by τ re-selection noise.

Because τ is re-selected on the recalibrated validation scores, part of the retained gain may be re-selection repairing calibrator damage rather than a component untouched by calibration; on TabICL and TabFM the forced-Dirichlet gains even *exceed* the native ones ($+4.62$ vs. $+3.21$; $+4.15$ vs. $+3.05$) while the calibrator lowers native macro-F1.

F Class selection across model families and checkpoints

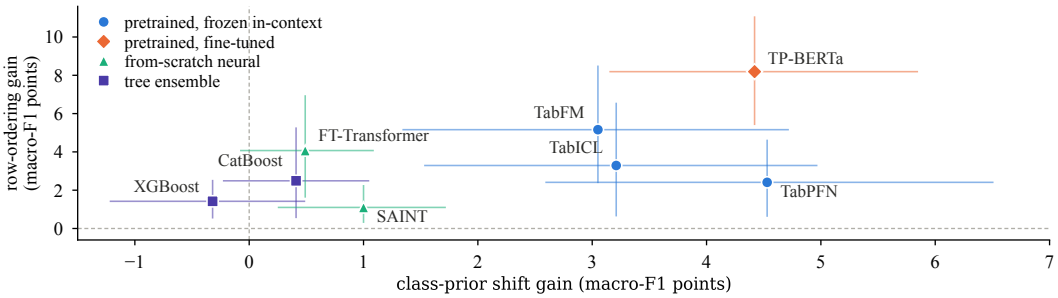

Figure 5: **Class decisions and row ranking vary across model families.** Incremental ridge gain against class-prior shift gain, macro-F1 points with 95% dataset-bootstrap intervals; one point per family on its own low-performing datasets (n=76/77/74 TabPFN/TabICL/TabFM, 121 TP-BERTa, 78 XGBoost, 77 CatBoost, 61 FT-Transformer, 94 SAINT), so points are not paired. TP-BERTa's row-ordering mean is inflated by a few small-test datasets (Tables 24, 21; Appendix F).

On datasets where both groups are low-performing, pretrained models leave $+1.97\,[+1.13, +2.81]$ macro-F1 more of it (Appendix F.1). A controlled intervention on a synthetic pretraining prior moves the component in the predicted direction ($+3.09\,[+0.31, +5.99]$ under the shipped rule against a balanced one, Appendix F.3), and both errors persist in a newer checkpoint (Appendix F.4). The subsections below give the full cross-family evidence summarized in Section 4: a matched comparison on datasets where both groups are low-performing, robustness checks, the controlled pretraining-prior intervention, and the newer checkpoint.

These analyses address dataset composition and model strength but do not isolate pretraining from architecture, training objective and adaptation procedure, so the cross-family result is an association with the pretrained models we evaluate. The intervention trains small models on the released TabICL v1 generator, not the released checkpoints, so it is not evidence about them.

F.1 Matched comparisons across model families

On datasets where both groups are low-performing, the estimated difference is $+1.97\,[+1.13, +2.81]$ macro-F1 with dataset fixed effects, $+2.78\,[+1.95, +3.69]$ pooled, and the gap survives AUC and log-loss selection rules. Applying the same decomposition to each family on its own low-performing datasets separates the two components sharply (Table 24). The class-prior shift gain is $+3.05$ to $+4.53$ for the three frozen in-context models but -0.32 to $+0.49$ for XGBoost, CatBoost, and the from-scratch FT-Transformer (five training seeds, all statistically zero), while across all eight families the ridge residual correction improves macro-F1 by $+1.10$ to $+8.19$ points, with a positive ranking-AUC point estimate in every family and an interval excluding zero in seven of eight (SAINT is inconclusive). Standard TabPFN fine-tuning retains a large class-selection component ($+6.81$) whereas class-balanced fine-tuning substantially reduces it ($+1.21$; cross-checked in Appendix J). Two further families show the same association: pretrained TP-BERTa, fine-tuned per dataset, retains $+4.42\,[+3.15, +5.85]$, whereas from-scratch SAINT leaves $+1.00\,[+0.25, +1.72]$, matching the trees and the FT-Transformer. We draw no conclusion about SAINT's row-ordering error, whose ranking-AUC gain ($+0.62\,[-0.05, +1.43]$) does not exclude zero and whose macro-F1 row gain is pipeline-sensitive. A minimum-test-size check ($\geq$30 test rows) preserves the class-selection pattern and shows TP-BERTa's large row-correction figure to be partly inflated by a few small-test datasets, whereas its ranking-AUC gain, the boundary-crossing evidence, survives the filter ($+5.79\,[+4.27, +7.60]$ unfiltered, $+4.70\,[+3.07, +6.54]$ filtered). Class selection is thus larger in the pretrained models of this comparison, while row-ordering error is more broadly present; the effect is not unique to in-context learning, since logistic regression and LDA can also leave substantial class-selection gains (Appendix K).

Matched pretrained-versus-from-scratch comparison. Because each family is evaluated on its own low-performing population, we also compare the two groups on shared datasets. Across the 101 datasets where at least one family from each group is low-performing (596 family–dataset observations), a regression of per-dataset class-prior shift gains on a pretrained indicator with dataset fixed effects and native macro-F1 as a control gives a pretrained coefficient of $+1.97\,[+1.13, +2.81]$ macro-F1 points, the interval bootstrapping datasets as the clustering unit; adding log number of classes and log training size in a pooled specification gives $+2.78\,[+1.95, +3.69]$ over 676 observations. Both fits come from one run, and dataset fixed effects are identified only by datasets carrying both a pretrained and a from-scratch family. The corresponding raw shared-dataset means are +3.20 for pretrained models and +0.88 for from-scratch models.

Decision-rule-free gates. The low-performing gate is validation macro-F1 < 0.60, the metric the class-prior shift repairs. To check that the frozen-minus-tree gap is not an artifact of that choice, we re-select per family the same number of datasets (78/79/87/78/78 for TabPFN/TabICL/TabFM/XGBoost/CatBoost; seed-42 score-only caches, $\leq$8K rows, $C \geq 2$) by lowest validation ranking AUC or highest validation log-loss, and compare on the datasets gated for all five families: the paired gap and the frozen coefficient of a dataset-fixed-effects regression controlling for validation AUC both have lower bounds above zero under all three gates (Table 23).

Table 23: Frozen-minus-tree class-prior shift gap under three gates of matched per-family size (datasets gated for all five families, from seed-42 score-only sets, so the macro-F1 row is not the +3.05 gap of Appendix E.3; macro-F1 points; 95% dataset-bootstrap intervals, B=10^4). "FE | val AUC" is the frozen coefficient with dataset fixed effects, controlling for validation ranking AUC (260/230/235 observations).

gate (per family, matched size)	n shared	paired gap	FE | val AUC
validation macro-F1 < 0.60 (fixed)	52	$+2.47\,[+1.00, +4.03]$	$+2.78\,[+1.22, +4.43]$
validation ranking AUC (lowest)	46	$+2.95\,[+1.44, +4.61]$	$+3.44\,[+1.93, +5.09]$
validation log-loss (highest)	47	$+2.12\,[+0.40, +3.90]$	$+2.60\,[+0.84, +4.42]$

Robustness to family composition and posterior flatness. The pretrained coefficient remains positive in every leave-one-family-out fit (+1.73 to +2.29), so no single family drives the result. Flatter predicted probability distributions do not explain it either: adding mean normalized posterior entropy to the regression leaves the estimate essentially unchanged ($+2.45\,[+1.63, +3.34]$ without it and $+2.51\,[+1.49, +3.59]$ with it), entropy is not a significant predictor, the from-scratch FT-Transformer has similarly diffuse predictions but a much smaller class-selection component, and per-dataset posterior sharpness is uncorrelated with the class-prior shift gain when pooled ($|\rho| \leq 0.07$; $n = 458$ family–datasets). A global temperature cannot enter the comparison at all, since $\arg\max_c(z_c/T + \tau \log \pi_c) = \arg\max_c(z_c + T\tau \log \pi_c)$ and τ is re-selected on validation. The larger class-selection gain of the pretrained models is therefore associated with the *ordering* of their class scores, not with overall posterior confidence; the ordering property that statistically absorbs it is hedging (Appendix E.1). Two scale-free descriptors of that ordering also fail to explain the gap, moving the pretrained coefficient only from +2.84 to +2.58. Miscalibration, posterior flatness and those descriptors therefore all fail to account for the difference, whereas the hedging statistic statistically absorbs it in a same-sample regression.

The same decision-rule phenomenon appears in fine-tuned TP-BERTa: under the self-consistency test of Appendix E.3 it reproduces 71% [48%, 111%] of its observed class-selection gain ($n = 136$), matching the frozen in-context models, so the large component left by pretrained models is consistent with a decision-rule effect whether the model is frozen or fine-tuned. SAINT's component is too small for a stable ratio (null +0.85 versus observed +0.86; $n = 104$).

Population and scoring notes. Directly trained baselines are averaged over five training seeds on the fixed seed-42 populations; frozen-model rows are deterministic given the cached probabilities, and the fine-tuned rows are single realizations cross-checked by the five-seed analysis of Appendix J. Populations differ from Table 15 and Section 3 because this

Table 24: **Two-component decomposition across model families**, each on *its own* low-performing datasets (validation macro-F1 < 0.60). "Freq. Shift" (class-prior shift) reports point estimates; "Row Corr." (ridge residual correction) reports 95% intervals in its macro-F1 and ranking-AUC columns. Higher values indicate more recoverable error.

model family (own low-performing set)	n	Freq. Shift gain	+ Row Corr. gain	+ Row Corr. AUC gain
TabPFN (frozen)	76	+4.53	+2.41 [+0.61, +4.64]	+2.10 [+0.42, +4.23]
TabICL (frozen)	77	+3.21	+3.29 [+0.63, +6.57]	+3.58 [+0.73, +7.31]
TabFM (frozen)	74	+3.05	+5.16 [+2.37, +8.51]	+4.30 [+2.08, +6.97]
TP-BERTa (pretrained, fine-tuned)	121	+4.42	+8.19 [+5.40, +11.08]	+5.79 [+4.27, +7.60]
XGBoost	78	−0.32	+1.42 [+0.52, +2.54]	+2.48 [+0.73, +4.82]
CatBoost	77	+0.41	+2.49 [+0.54, +5.28]	+2.97 [+0.87, +5.65]
FT-Transformer (from scratch)	61	+0.49	+4.07 [+1.61, +6.96]	+4.40 [+1.70, +7.38]
SAINT (from scratch)	94	+1.00	+1.10 [+0.29, +2.27]	+0.62 [−0.05, +1.43]
TabPFN, standard fine-tune	61	+6.81	+4.28 [+1.70, +7.53]	+2.43 [+0.75, +4.50]
TabPFN, class-balanced fine-tune	62	+1.21	+2.67 [+0.70, +4.97]	+2.23 [+0.79, +4.02]

analysis uses each family's own ≤8K low-performing set; for the three frozen models these are the same 76/77/74 datasets as the main decomposition, whose uncorrected macro-F1 values (46.9/48.3/46.8 against 46.26/48.24/46.36) differ through the scoring convention (Appendix C), not a different population. TabICL's ranking-AUC value uses the 75 datasets with archived AUC-track predictions (Table 13).

These analyses address dataset composition and model strength, but do not isolate pretraining from architecture, training objective and adaptation procedure. We therefore interpret the result as an association with the pretrained models we evaluate (the estimate varying across data splits for TabICL and TabFM; Appendix G), not as a causal effect of pretraining.

F.2 Covariate controls and tuned tree baselines

With full controls, the in-context coefficient is +3.05 [+2.05, +4.02], and tuned trees still reach only +1.14 and +0.07. The cross-family difference is not explained by weaker initial performance or dataset composition. Regressing per-dataset class-prior shift gain on model-family indicators, controlling for validation macro-F1, class imbalance, training-set size, number of classes and benchmark suite, gives an in-context coefficient of +3.05 [+2.05, +4.02] on the ≤20K population, similar with binned performance controls (+3.10), leave-one-suite-out fits (+2.89 to +3.29), and on datasets shared across all families (+3.38). Family-level analyses agree, with larger uncertainty given only a few families.[2]

Tuning the tree baselines does not remove the difference: after a 20-trial per-dataset search (validation-macro-F1 objective, five training seeds) the class-prior shift gains are +1.14 [+0.26, +2.07] (XGBoost) and +0.07 [−0.87, +0.96] (CatBoost), still below the frozen range of +3.05 to +4.53, and the covariate-adjusted in-context differential remains +2.40 [+1.41, +3.34]. Stronger XGBoost tuning narrows the gap somewhat, but the difference is not a default-hyperparameter artifact.

Folding logistic regression and LDA in as their own groups places the in-context indicator at +2.95 [+1.92, +4.02], above the tree and from-scratch reference, but LDA (+2.84 [+1.60, +4.09]) is indistinguishable from it and logistic regression (+1.77 [+0.21, +3.31]) comparable, so the claim is one of *degree relative to the task-trained trees and from-scratch neural models*, not uniqueness (Appendix K).

F.3 Pretraining class-imbalance intervention

In small models of this architecture, training on the released prior rather than a balanced one leaves a class-prior shift gain +3.09 [+0.31, +5.99] macro-F1 points larger and a losing share +0.143 [+0.081, +0.210] higher. This does not establish that pretraining imbalance causes hedging in the released checkpoints.

[2]Five-seed family-level battery: wild-cluster bootstrap imposed-null $p = 0.019$ ($G = 8$); crossed family/dataset random-effects in-context coefficient +2.94; three-versus-three family-means $t = 5.22$. The ≤8K population gives the same picture (+2.99 [+1.94, +3.99]; family wild-cluster $p = 0.020$).

The hedging statistic above describes posteriors; it does not say where the hedging comes from. The pretraining data of the checkpoints we study are not public, so we intervene on a released synthetic prior instead. At its default settings the released TabICL v1 generator produces label distributions far more skewed than our datasets (median largest-to-smallest class ratio 8.0 against 1.9 over our 361 datasets with at most 8K rows); the v2 prior of the checkpoint we evaluate is not released, so this describes the v1 generator only. We trained three 27.7M-parameter TabICL-architecture models from scratch with that prior and its own trainer, identical in architecture, trainer configuration, steps and data seeds, differing *only* in the rule that sets class proportions: BALANCED places class boundaries at equal quantiles, DEFAULT leaves the shipped rule unchanged, and LONGTAIL draws class proportions from a Dirichlet(0.3). The realized dose, measured on 2048 synthetic datasets per arm before training, is a mean minority share of 0.356/0.150/0.099 and a geometric-mean class ratio of 1.005/8.76/25.2: ordered as registered, but unevenly spaced. Each arm was trained with two torch seeds for 7,000 steps at batch 128 and a 1,024-row training length, and evaluated on a population fixed before training: the 72 of TabICL's 77 primary low-performing datasets that fit the small models' limits, each dataset's context capped at the trained length by the same stratified subsample in every arm.

Table 25: **Small TabICL-architecture models trained on priors that differ only in class imbalance.** Means over the 72 evaluation datasets, seed-averaged within an arm (two training seeds per arm); macro-F1 and ranking AUC ×100, shift gain in macro-F1 points. "prior IR" is the geometric-mean largest-to-smallest class ratio of the arm's training prior, measured on 2048 synthetic datasets before training. The predicted minority share is followed by the mean over datasets of the per-dataset ratio of predicted to observed minority share, not the quotient of the two printed means. Reference lines are the seed-42 tree caches on the same 72 datasets; the released TabICL checkpoint is descriptive, not an arm of this intervention.

arm (prior's class-proportion rule)	prior IR	native macro-F1	ranking AUC	losing share	shift gain	pred. minority share	median τ
BALANCED (equal quantiles)	1.005	46.58	68.51	0.611	+1.15	0.417 (0.93)	+0.05
DEFAULT (shipped rule)	8.76	46.18	69.94	0.754	+4.24	0.200 (0.43)	−0.66
LONGTAIL (Dirichlet(0.3))	25.2	45.73	69.22	0.769	+4.06	0.192 (0.41)	−0.65
XGBoost (reference)	–	51.99	68.65	0.460	−0.13	0.318	–
CatBoost (reference)	–	50.13	68.36	0.522	+1.52	0.299	–
released TabICL (descriptive)	–	48.97	70.89	0.684	+3.48	0.255	–

On held-out synthetic tasks, the manipulation check separates the arms: on held-out synthetic tasks with a fixed imbalanced class distribution (true minority share 25.1%), the pooled share of rows predicted as a minority class is 6.26% for BALANCED, 4.26% for DEFAULT and 4.10% for LONGTAIL, meeting the pre-registered separation of at least 2 percentage points (2.16 pp). The registered quality gates also pass: every arm reaches at least 44 native test macro-F1 (minimum 45.73), and the arms differ by 0.86 macro-F1 and 1.43 ranking-AUC points, inside the registered tolerances of 3 and 2 points. Table 25 reports the arms. Hedging rises with the prior's class imbalance: the losing share increases by $+0.079\,[+0.049, +0.111]$ per dose step, an interval stable across bootstrap seeds. The claim-bearing contrast is the pairwise BALANCED-versus-DEFAULT comparison: the class-prior shift gain is $+3.09\,[+0.31, +5.99]$ macro-F1 points larger under the shipped prior, with a $+0.143\,[+0.081, +0.210]$ larger losing share and a $-0.217\,[-0.296, -0.144]$ lower predicted minority share. The registered ranking-AUC dose slope covers zero ($+0.36\,[-0.10, +0.81]$ points per step, well inside the registered tolerance of 2 points).[3]

Three limits bound what this experiment supports. *The dose is not graded.* DEFAULT and LONGTAIL do not separate: they are indistinguishable in the manipulation check (4.26 vs. 4.10 pp) and their contrasts cover zero (losing share $+0.015\,[-0.001, +0.033]$, whose lower bound is Monte-Carlo unstable across bootstrap seeds in the same way as the dose slope of the footnote above; gain $-0.18\,[-1.59, +1.23]$), so the evidence rests on the single BALANCED-versus-DEFAULT contrast. *Imbalance is not separated from model quality.* BALANCED leads on native macro-F1 by 0.86 and has the smallest gain, which is what a pure strength difference would also produce, while DEFAULT ranks $+1.43\,[+0.43, +2.45]$ ranking-AUC

[3]We report the pairwise contrast rather than the registered per-dose slope of the gain, whose bootstrap lower bound is Monte-Carlo unstable. The pre-registered single-member sensitivity is uninterpretable under the registered rule, because LONGTAIL falls below the 44 macro-F1 quality bar there.

points above it; the spread is inside the registered tolerance, the macro-F1 slope covers zero ($-0.43\,[-1.38, +0.52]$), and the ranking direction is favourable to the reading here, since the arm with the larger class-selection gain is the better-ranking one, but weaker models do tend to have larger class-selection gains and we cannot separate a quality difference from the effect. *These are not the released checkpoints.* They are small models trained for 7,000 steps on the released TabICL v1 generator with the v1 trainer and a capped context, so the experiment shows only that the class imbalance of this synthetic prior changes hedging and the class-selection gain in models of this architecture trained this way. It is not evidence about the released checkpoints, whose own prior is unreleased, and it does not establish that pretraining causes hedging in general.

F.4 Decision and ranking gains on TabPFN-3.5

Both gains persist on TabPFN-3.5. It is $+2.28\,[+0.44, +4.08]$ macro-F1 stronger than v3 natively, yet the class-prior shift still recovers $+3.06\,[+1.49, +4.62]$ and the ridge residual correction a further $+2.94\,[+0.83, +5.48]$.

We evaluate TabPFN-3.5 using `tabpfn` 9.0.0, which sets it as the default checkpoint, a checkpoint-declared softmax temperature (1.0, against 0.9 for v3), and new decision-threshold options. We compare v3 and v3.5 on the same 76 datasets selected using v3 validation predictions (the fixed set of Table 13), under frozen seed-42 inference with the splits, preprocessing and uncapped context of the main pipeline, in four settings: cached v3 predictions (re-running v3 in the new environment reproduces the cache to $\max|\Delta p| = 5.4 \times 10^{-5}$ with argmax agreement 1.0000, so the library upgrade does not confound the comparison); v3.5 at library defaults; v3.5 with `balance_probabilities=True`; and v3.5 with its own per-class decision-threshold tuner (`ClassifierTuningConfig`, `eval_metric="f1"`). Because a stronger model leaves fewer low-performing datasets, we also re-applied the <0.60 validation gate to v3.5's own predictions over the same candidate pool, giving a reselected set with $n = 73$ (overlap 69 datasets). We use a paired dataset-clustered bootstrap (10^4 replicates) for the fixed-set comparison; the two decision-option comparisons are descriptive.

Table 26: **Both components persist on TabPFN-3.5.** Macro-F1 $\times 100$ and gains in macro-F1 points, dataset-clustered 95% bootstrap intervals (10^4 replicates, one shared resample index per replicate). The fixed set uses v3 validation selection ($n = 76$); the reselected set re-applies the <0.60 low-performing rule to v3.5's own validation predictions ($n = 73$, its ridge correction and total columns resting on the 71 datasets where ridge correction is defined). The ridge correction gain is incremental over class-prior shift and the total applies both. Intervals come from this run's own bootstrap; point estimates match Table 14.

population	condition	initial macro-F1	ranking AUC	shift gain	ridge gain	total gain
fixed (n=76)	v3 (cached)	46.26 [42.22, 50.39]	71.08	+4.53 [+2.60, +6.43]	+2.41 [+0.61, +4.65]	+6.93 [+4.32, +9.83]
	v3.5 defaults	48.53 [44.78, 52.39]	72.16	+3.06 [+1.49, +4.62]	+2.94 [+0.83, +5.48]	+6.00 [+3.42, +8.81]
	v3.5, `balance_probabilities`	48.46 [44.51, 52.47]	71.70	+3.04 [+1.45, +4.69]	+2.87 [+0.73, +5.41]	+5.90 [+3.34, +8.71]
	v3.5, threshold tuner	46.41 [42.66, 50.20]	71.93	+4.26 [+2.57, +5.97]	+3.89 [+1.55, +6.58]	+8.15 [+5.09, +11.52]
reselected (n=73)	v3 (cached)	45.60 [41.39, 49.94]	71.45	+4.79 [+2.74, +6.83]	+2.58 [+0.68, +4.92]	+7.21 [+4.42, +10.21]
	v3.5 defaults	47.69 [43.67, 51.69]	72.05	+3.45 [+1.82, +5.10]	+3.00 [+0.79, +5.70]	+6.21 [+3.50, +9.25]

On the fixed set in Table 26, the paired change in the shift gain is not resolved ($-1.47\,[-3.25, +0.32]$), the row gain if anything grows ($+0.53\,[-0.93, +2.39]$), the combined gain is indistinguishable ($+6.00$ against $+6.93$), and the share of datasets on which the deployed component loses is essentially unchanged. The same pattern holds on the reselected set and its intersection with the fixed set.

TabPFN offers two additional decision options; the tested settings leave both correction gains measurable. Enabling `balance_probabilities` leaves a tuned shift on top of it recovering $+3.04\,[+1.45, +4.69]$: the flag over-corrects rather than capturing the component, moving the selected τ from negative (mean -0.58) to positive (mean $+0.36$, positive on 75% of datasets) so that the tuner pulls back. The released per-class decision-threshold tuner fits thresholds on a holdout built from the training data and divides the probabilities by them before the argmax, placing it in the same functional family as the class-prior shift. On this low-performing population its native macro-F1 reads $-2.12\,[-4.51, +0.17]$ against default v3.5 and $+0.15\,[-2.39, +2.68]$ against v3; a tuned shift on top still gains $+4.26\,[+2.57, +5.97]$, and the row correction gains more after it than before ($+3.89$ against

+2.94; paired +0.95 [+0.04, +2.08]), giving the largest total gain of any arm. Both components therefore remain measurable after the shipped machinery has been applied.

v3 and v3.5 ship different softmax temperatures. Re-scoring each cached posterior at the other temperature in closed form and re-selecting τ moves the population-mean shift gain by only +0.33 [−0.15, +0.86] for v3 and +0.50 [−0.07, +1.16] for v3.5, both intervals covering zero, so the temperature difference does not account for the observed gains. Per-dataset variation under that re-scoring is much larger.

G Robustness to dataset size, class count, and split

Size and class count leave both components in place. Across alternate splits, the result is *split-sensitive*: the tuned class-selection interval covers zero on two alternates, while the row-ordering interval excludes zero on every alternate for all three backbones. Table 27 compares the primary $\leq$ 8K decomposition with a $\leq$ 20K re-run on each model's low-performing datasets; both caps count total rows across the train, validation, and test splits (Section 4), and raising the cap yields similar total and component gains. The primary analysis uses $\leq$8K because every dataset then lies within each model's ordinary in-context operating range (Appendix C); the cap fixes the models' in-context behavior and is not a claim that the effect exists only on small data. On the $\leq$ 20K populations the class-prior shift component is +4.79 [+2.98, +6.66] (TabPFN), +3.45 [+1.79, +5.15] (TabICL), and +3.33 [+1.71, +4.90] (TabFM); Table 34 gives the row-correction and ranking-AUC intervals on the same populations (five seeds, dataset bootstrap). Both components remain significant on every model.

Table 27: Sensitivity to the dataset-size cap (total rows across train, validation and test splits). Raising the cap from 8K to 20K leaves the total and component gains similar on all three models. Point estimates in macro-F1 points; higher is better.

model	cap	n	flexible benchmark	Freq. Shift	+ Row Corr.	ranking AUC
TabPFN	$\leq$8K	76	7.65	4.53	2.41	2.10
	$\leq$20K	81	7.05	4.79	2.26	1.95
TabICL	$\leq$8K	77	5.73	3.21	3.29	3.58
	$\leq$20K	80	5.91	3.45	3.14	3.43
TabFM	$\leq$8K	74	9.71	3.05	5.16	4.30
	$\leq$20K	79	9.28	3.33	4.89	4.10

Binary versus multiclass datasets. The primary populations are mostly binary (Section 4; on their score-only supersets, median training size 400–500 rows, median two classes, 54–62% binary, median imbalance ratio 2.2–3.7). Splitting both components by class count (Table 28), the class-selection component is concentrated on binary tasks, where the shift is exactly one-parameter threshold tuning, and stays small for both trees on either subset. The row-ordering increment has no consistent pattern: larger on multiclass tasks for TabPFN, larger on binary tasks for TabICL, similar across the split for TabFM, and positive on binary tasks for both trees.

Alternate-split replication. A multi-split check on the primary TabPFN population (two alternate train/validation/test splits, fresh re-inference, the low-performing filter re-applied per split) gives intervals excluding zero for both components on both alternates (Table 29): the class-prior shift component has across-split mean $+4.26 \pm 0.25$ with the printed +4.53 re-inference anchor reproducing exactly, while the row component reaches +3.53 and +4.35, of which the printed +2.41 is the conservative end. Of the 76 datasets, 64 remain low-performing across all splits (six and nine cross above the 0.60 threshold on the alternates, none enter).

We apply the same split generator and selection criterion to the TabICL (77) and TabFM (74) populations; 62 and 59 datasets are low-performing on all three splits, and alternate test labels differ from the original split's on every dataset. Table 29 reports, per alternate split,

Table 28: **Both components by class count.** class-prior shift gain (class-prior shift) and ridge residual correction increment (ridge correction), macro-F1 track, on the macro-F1-track populations of Appendix G.1 ($n = 76/77/86/78/78$), split into binary and multiclass datasets: mean with 95% interval, median, and win/loss/tie counts (tie: $|\text{gain}| \leq 0.005$). The validation gate deploys the row correction on a minority of datasets, so the ridge correction median is zero even where its mean is significant.

			class-prior shift (class selection)			ridge correction (row ordering)		
family	subset	n	mean	median	W/L/T	mean	median	W/L/T
TabPFN	C=2	41	+6.33 [+3.48, +9.02]	+4.38	31/6/4	+1.06 [−0.56, +3.71]	0.00	4/5/32
	$C\geq$3	35	+2.41 [+0.06, +5.02]	+1.28	20/13/2	+3.99 [+1.07, +7.58]	0.00	12/1/22
TabICL	C=2	43	+4.65 [+2.36, +7.03]	+2.86	29/6/8	+4.81 [+0.45, +10.18]	0.00	7/9/27
	$C\geq$3	34	+1.40 [−1.02, +3.71]	+0.05	17/10/7	+1.37 [−0.07, +3.70]	0.00	5/2/27
TabFM	C=2	53	+3.67 [+0.85, +6.76]	+0.85	28/15/10	+5.32 [+1.87, +9.45]	0.00	14/7/32
	$C\geq$3	33	+1.10 [−1.49, +3.50]	+2.22	19/11/3	+5.76 [+1.69, +10.62]	0.00	11/2/20
XGBoost	C=2	42	+0.62 [−0.91, +2.02]	0.00	15/15/12	+1.89 [+0.42, +3.74]	0.00	7/2/33
	$C\geq$3	36	−1.06 [−2.42, +0.13]	0.00	10/13/13	+1.04 [−0.51, +2.74]	0.00	9/3/24
CatBoost	C=2	41	+1.03 [+0.05, +2.11]	0.00	14/12/15	+3.64 [+0.15, +8.53]	0.00	6/2/33
	$C\geq$3	37	+1.14 [−0.22, +2.43]	+0.35	20/9/8	+0.83 [−0.68, +2.69]	0.00	6/8/23

the validation-tuned class-prior shift gain, the gain with τ selected without validation labels (posterior selection, Appendix E.2), and the ridge residual correction gain. The posterior selection interval excludes zero on all four TabICL and TabFM alternates. Re-inference on the original partitions gives tuned shift and row gains of +3.20 and +3.29 (TabICL, n=77) and +3.12 and +5.00 (TabFM, n=73; TabFM re-inference is not bit-exact).

Table 29: **Alternate-split replication.** Macro-F1 gains on each backbone's low-performing datasets under two alternate splits (fresh re-inference, gate re-applied per split), with 95% dataset-bootstrap intervals (10^4 replicates). Tuned: τ selected on validation labels; posterior selection: τ from the model's own validation posterior (Appendix E.2; not run for TabPFN); row: ridge residual correction. Pooled rows are descriptive, over dataset–split units from both alternates (71 and 70 datasets), dataset-clustered.

backbone	split	n	class selection, tuned	class selection, posterior	row ordering
TabPFN	101	70	+4.09 [+2.46, +5.98]	—	+3.53 [+1.24, +6.18]
TabPFN	202	67	+4.14 [+1.99, +6.30]	—	+4.35 [+1.22, +7.82]
TabICL	101	69	+2.61 [+0.91, +4.38]	+3.77 [+2.02, +5.79]	+4.22 [+1.23, +8.00]
TabICL	202	64	+1.79 [−0.64, +4.20]	+2.87 [+0.86, +4.94]	+3.51 [+0.45, +7.35]
TabICL	pooled	133	+2.22 [+0.69, +3.77]	+3.33 [+1.77, +4.94]	—
TabFM	101	66	+0.39 [−1.23, +1.97]	+2.74 [+1.52, +4.04]	+6.81 [+3.54, +10.55]
TabFM	202	63	+3.16 [+1.50, +4.88]	+3.72 [+2.27, +5.21]	+5.50 [+1.96, +9.60]
TabFM	pooled	129	+1.75 [+0.41, +3.08]	+3.22 [+2.12, +4.39]	—

The effect is *split-sensitive*: the tuned class-selection interval covers zero on TabICL split 202 and TabFM split 101, whereas the row-ordering interval excludes zero on every alternate split for all three backbones.

G.1 Class selection under balanced accuracy

Under balanced accuracy, each pretrained backbone retains a larger shift gain than either tree, and the row-ordering component barely moves. Macro-F1's optimal decision rule is a threshold rule (Lipton et al., 2014; Narasimhan et al., 2015), so the class-prior shift is a restriction of, not an instance of, the optimal class-wide family, whereas balanced accuracy is decomposable and its Bayes-optimal rule is exactly prior-reweighted argmax, $\arg\max_c p_c(x)/\pi_c$, the shift at $\tau = -1$. We therefore rerun the native, class-prior shift, and general per-class bias arms with the selection metric switched to balanced accuracy (Table 30), on each family's own low-performing population with the 8K-row cap and featurization filter, recomputing the <0.60 validation gate from this run's own validation predictions, which gives $n = 76/77/86/78/78$ on the macro-F1 track and 76/75/84/76/76 on the balanced-accuracy track. The macro-F1 column matches the main-text values for TabPFN and TabICL on

the same datasets; for TabFM, on 86 rather than the primary 74 datasets, the value moves from +3.05 to +2.69. The selection metric is therefore the operative difference between the columns, and the population reconciliation is in Table 10.

Table 30: **Both components survive a change of objective.** class-prior shift gain in macro-F1 (f1 track) and macro-recall (balanced-accuracy track), then the increments from a general per-class bias and from the ridge residual correction. Dataset-bootstrap 95% intervals; $n = 76/77/86/78/78$ and 76/75/84/76/76 (TabPFN/TabICL/TabFM/XGBoost/CatBoost).

family	shift gain, bal. acc.	shift gain, macro-F1
TabPFN	+4.27 [+2.10, +6.65]	+4.53 [+2.62, +6.49]
TabICL	+4.82 [+2.51, +7.33]	+3.21 [+1.49, +4.93]
TabFM	+3.96 [+2.08, +6.10]	+2.69 [+0.66, +4.78]
XGBoost	+2.05 [+0.61, +3.65]	−0.15 [−1.19, +0.83]
CatBoost	+2.42 [+1.06, +3.83]	+1.08 [+0.25, +1.91]
general per-class bias, increment over the shift		
TabPFN	−0.90 [−2.66, +0.69]	−1.70 [−3.21, −0.14]
TabICL	−0.34 [−1.84, +1.09]	−2.40 [−4.10, −0.90]
TabFM	−0.28 [−1.70, +1.45]	−2.09 [−3.73, −0.55]
ridge residual correction, increment over the shift		
TabPFN	+2.22 [+0.17, +4.81]	+2.41 [+0.64, +4.59]
TabICL	+3.15 [+0.95, +5.87]	+3.29 [+0.74, +6.67]
TabFM	+5.48 [+2.68, +8.59]	+5.49 [+2.74, +8.59]
XGBoost	+1.74 [+0.60, +3.08]	+1.50 [+0.38, +2.74]
CatBoost	+3.28 [+0.99, +6.30]	+2.31 [+0.33, +5.09]

Four observations. First, every pretrained backbone remains above both trees under balanced accuracy (+3.96 at worst against +2.42 at best), so macro-F1 does not produce the cross-family difference. Second, both trees are significantly positive here, whereas under macro-F1 XGBoost is null and CatBoost is null on its primary population (Table 21) and only marginally positive on this one, which is why we describe the component as larger in the pretrained models rather than absent elsewhere. Third, the general per-class bias significantly *loses* to the one-parameter shift under macro-F1 on all three backbones but ties it under balanced accuracy, where the class-wide family is correctly specified: the advantage of the restricted form is estimation variance on small validation sets, not evidence that the error is one-dimensional. Consistently, the selected τ is more negative under balanced accuracy in every family, with TabPFN's median at -1.05, close to the $\tau = -1$ theory predicts when the posterior is correct. Fourth, the row-ordering component is essentially unchanged by the switch, as a genuine ranking property should be: every family gains under both objectives with intervals excluding zero, and the point estimates barely move (+5.48 against +5.49 on TabFM, +2.22 against +2.41 on TabPFN).

Balanced accuracy also has a non-argmax Bayes rule, so this cannot show that the class-selection component is independent of the decision rule. The margin narrows and the TabFM and CatBoost intervals overlap, so the sharper statement still rests on the matched within-dataset comparison of Appendix F.1.

H Adaptation across backbone performance levels

Class selection vanishes first. Below 0.60 both components are large; in $[0.60, 0.80)$ the shift falls to +0.66 [−0.41, +1.75] while the row correction still adds +1.44 [+0.47, +2.60] macro-F1 and +0.89 [+0.10, +1.81] ranking AUC; above 0.80 both are negligible. Table 31 reports the class-prior shift macro-F1 gain by band of before-correction validation macro-F1, with the mean minority pred/obs, over all datasets rather than only low-performing ones. The gain falls toward zero as before-correction macro-F1 rises, monotonically for TabPFN and with small fluctuations for TabICL and TabFM, while minority selection moves from severe underselection (pred/obs $\approx$ 0.3–0.5) to near-parity ($\approx$ 0.97) on the strongest slice: a property of the low-performing regime, not an artifact of the 0.60 cutoff.

Table 31: class-prior shift macro-F1 gain by before-correction macro-F1 band (test, 95% intervals), with mean minority pred/obs. The gap vanishes as models strengthen.

before-correction macro-F1 band	TabPFN gain (pred/obs)	TabICL gain (pred/obs)	TabFM gain (pred/obs)
< 0.50	$+6.02\,[+3.48, +8.48]$ (0.26)	$+4.03\,[+2.27, +5.92]$ (0.47)	$+4.25\,[+2.20, +6.24]$ (0.42)
$[0.50, 0.60)$	$+2.56\,[+0.99, +4.13]$ (0.60)	$+2.40\,[-0.97, +5.51]$ (0.68)	$+1.65\,[-0.78, +4.25]$ (0.67)
$[0.60, 0.70)$	$+0.67\,[-1.03, +2.44]$ (0.70)	$+2.21\,[+0.59, +3.87]$ (0.67)	$+1.77\,[+0.02, +3.62]$ (0.71)
$[0.70, 0.80)$	$+0.20\,[-1.60, +1.85]$ (0.80)	$-0.11\,[-1.19, +0.84]$ (0.81)	$+0.71\,[-0.84, +2.22]$ (0.84)
≥ 0.80	$-0.07\,[-0.24, +0.09]$ (0.97)	$+0.23\,[-0.19, +0.79]$ (0.96)	$+0.05\,[-0.37, +0.44]$ (0.97)

Row ordering and transfer across the spectrum. The same banding applied to the row-ordering component, pooled across the three in-context models (dataset-clustered 95% intervals; Section 4.3), shows it persisting one regime beyond class selection. In the <0.60 band the shift adds $+3.41\,[+2.14, +4.73]$ macro-F1 and the row correction $+3.75\,[+1.80, +6.03]$ (ranking AUC $+3.48\,[+1.56, +5.85]$); in $[0.60, 0.80)$ the shift falls to $+0.66\,[-0.41, +1.75]$ while the row correction still adds $+1.44\,[+0.47, +2.60]$ (AUC $+0.89\,[+0.10, +1.81]$); above 0.80 both are negligible ($+0.00\,[-0.27, +0.30]$ and $+0.23\,[+0.05, +0.47]$). Transfer persists similarly: a row correction fitted on a different in-context model recovers $+1.15\,[+0.21, +2.33]$ macro-F1 on an intermediate-band target, $\approx$78% of the target's own $+1.48\,[+0.48, +2.70]$. The corrections are the cached directions of Appendix H.3, evaluated on the splits the three models share.

The row-ordering repair is learnable from the input features alone (eight families). Table 32 compares, on each family's low-performing datasets, the ridge residual correction fit against the model residual ($\mathbf{1}[y{=}c] - p_{\text{model}}$) with an identical correction fit against the class-prior residual ($\mathbf{1}[y{=}c] - \pi_c$), which never observes the model's predictions. The two agree closely in aggregate (63–114% per family; paired difference $+0.05\,[-0.20, +0.29]$ macro-F1), so most of the repair is learnable from the input features alone.

Table 32: Row-ordering macro-F1 gain, model-residual ("own") versus a feature-only correction fit against the class-prior residual ("task-only"), by family on each family's low-performing datasets (one run on the seed-42 cached predictions). The two agree closely in aggregate (63–114% per family). This experiment's uniform pipeline makes per-family estimates differ from Table 24's, most for SAINT.

	TabPFN	TabICL	TabFM	TP-BERTa	XGBoost	CatBoost	FT	SAINT
own (model residual)	+2.41	+3.33	+5.55	+8.07	+1.50	+2.31	+4.49	+2.51
task-only (prior)	+2.07	+3.41	+5.28	+7.94	+0.95	+2.41	+5.10	+2.74

Table 33: Staged contraction for the four pretrained families: class-prior shift (class selection) and ridge residual correction (row ordering) macro-F1 gains by native validation macro-F1 band (each family on its own datasets; one run on the seed-42 cached predictions). Class selection falls sharply by the intermediate band for all four and row ordering remains larger there in all four, contracting more slowly except for TabFM, where both fall by about three quarters. Both columns come from the run of Table 32, on the datasets in each band passing the featurization filter, so they are not the primary populations of Table 24.

	< 0.60		$[0.60, 0.80)$		≥ 0.80	
family	class-sel.	row-ord.	class-sel.	row-ord.	class-sel.	row-ord.
TabPFN	+4.64	+2.41	+0.17	+0.91	−0.10	+0.24
TabICL	+3.18	+3.33	+1.09	+2.07	+0.29	+0.38
TabFM	+2.79	+5.55	+0.63	+1.33	−0.22	+0.05
TP-BERTa	+5.11	+8.07	+1.19	+4.49	+0.22	+1.38

Averaging control. Table 3 gives the four arms and Figure 2 the band picture. To test whether the row correction is ordinary ensembling, we compare four validation-tuned arms on *every* dataset ($\leq$ 8K rows, featurizable), pooled over the three frozen in-context models: the frozen model alone; the task-only feature model alone (scores $\log \pi_c$ plus the ridge adjustment fit to $\mathbf{1}[y{=}c] - \pi_c$, λ selected for this standalone objective); a probability average of the two, weight on a 21-point validation grid; and class-prior shift followed by the task-only

feature model (validation-selected, class-prior shift-only fallback). By performance band (validation macro-F1 < 0.60 / $[0.60, 0.80)$ / ≥ 0.80; 236/226/493 model–dataset pairs over 99/99/187 datasets, dataset-clustered intervals), class-prior shift followed by the task-only feature model improves on the frozen model by $+7.16\,[+4.91, +9.69]$, $+2.13\,[+0.63, +3.74]$ and $+0.26\,[-0.06, +0.61]$, while the weighted probability average gives $+3.95\,[+0.97, +7.46]$, $-0.53\,[-2.09, +1.19]$ and $-0.54\,[-1.36, +0.22]$. On the low-performing datasets the two learners are comparable and both arms help. In the intermediate band the frozen models are far stronger than the task-only feature model ($+17.2\,[+13.0, +21.5]$) and averaging adds nothing, yet that combination still adds macro-F1 within each backbone (TabPFN $+1.4\,[-0.3, +3.2]$, TabICL $+3.1\,[+0.9, +5.5]$, TabFM $+1.9\,[+0.2, +3.6]$) and $+0.8\,[+0.2, +1.7]$ ranking AUC; the pooled combination-minus-averaging contrast is $+2.66\,[+1.10, +4.37]$, per backbone $+1.65\,[-0.10, +3.58]$, $+4.07\,[+1.96, +6.60]$ and $+2.15\,[+0.19, +4.00]$, excluding zero for TabICL and TabFM. Above 0.80 every arm is within noise of the frozen model. The signal is therefore *complementary*: a far weaker task-only feature model still improves a much stronger backbone when its adjustment is added to the shifted scores, and averaging cannot extract that information — nor can a stacking meta-learner over the concatenated frozen and task-only log-probability vectors, which leaves that combination ahead by $+3.59\,[+1.32, +6.25]$ macro-F1 in the intermediate band.

Log-linear pooling and cross-fitted stacking. The class-prior shift followed by the task-only feature model is algebraically a log-linear pooling (Genest and Zidek, 1986; Hinton, 2002): with $q = \mathrm{softmax}(\log \pi + \Delta)$ the standalone task-only feature model, $\mathrm{softmax}(\log p + \tau \log \pi + \Delta) = \mathrm{softmax}(\log p + \log q + (\tau - 1) \log \pi)$. We therefore add the ensemblers that can express it, on the same rows and bands: a log-linear pool $\log p + \beta(\log q - \log \pi)$ with β on a validation grid, the same pool after the shift, a convex geometric pool, and logistic and boosted stackers on the cross-fitted *training* predictions. In the intermediate band the tuned log-linear pool ties that combination ($-0.19\,[-1.31, +0.92]$, mean $\beta = 0.40$) and the cross-fitted logistic stacker comes close, whereas the convex geometric pool and the boosted stacker do not; the tuned pool improves the frozen model by $+2.32\,[+1.08, +3.88]$, and the same holds on the low-performing band and for the task-trained models. Complementarity is therefore about the *information*, not the combination rule: a task-only feature model 17 points weaker than the backbone improves it when the two are pooled in log space, which probability averaging cannot express.

The same four arms on task-trained models. The identical protocol on XGBoost, CatBoost and the from-scratch FT-Transformer separates the two components of the intermediate-band result. The row-dependent increment over class-prior shift is not specific to the in-context models: in $[0.60, 0.80)$ it reaches $+0.72\,[+0.12, +1.57]$ for the task-trained models against $+1.49\,[+0.67, +2.48]$ for the frozen models. What differs is the class-wide part: the shift gains $+0.64\,[-0.47, +1.76]$ for the frozen models but loses $-0.75\,[-1.79, +0.09]$ for the task-trained ones, neither contrast separated from zero, consistent with committed posteriors leaving little to reallocate (Appendix E.1). The intermediate-band complementarity is therefore a property of the feature-range information rather than of in-context prediction.

TabFM is the exception to the staged contraction: both of its components fall by about three quarters by the intermediate band (Table 33). The transfer interval is wide and the effect is concentrated in a minority of datasets.

H.1 Separating row variation from class-wide means

The ridge gain reflects row-varying adjustments: the class-wide part is small and non-positive on two models, and re-centering recovers the full macro-F1 gain and the full AUC. Because the interval indicators are centered only on the training split, on test the fitted correction $\Delta(x) = \widetilde{\phi}(x)W$ can carry a nonzero per-class mean $m_c = \mathbb{E}_{\text{test}}[\Delta_c(x)]$, a class-wide constant invisible to one-vs-rest AUC (Fact 1) that could inflate the macro-F1 gain we attribute to row ordering. We decompose the correction on test into m plus a row-varying remainder and compare four test predictions (Table 34, on each model's low-performing datasets; higher macro-F1 is better): $A = z^{(1)}$ (class-prior shift only), $B = z^{(1)} + m$, $C = z^{(1)} + \Delta$

(the reported interval model), and $D = z^{(1)} + (\Delta - m)$ (test-re-centered, class-wide mean removed).

The class-wide component $(B - A)$ is small on every model and non-positive on two, so it contributes essentially nothing to the reported interval gain, and for TabPFN and TabFM slightly *subtracts* from it, because the class-prior shift already sits at its validation optimum. Re-centering the class-wide mean away $(D - A)$ recovers the full macro-F1 gain, and the AUC gain of D equals that of C to the reported digit, since removing a per-class constant cannot change AUC. The leftover mean $\|m\|$ is only 0.3–0.7% of the class-prior shift term and the global component accounts for under 4% of the correction's squared magnitude, so the row-ordering gain is genuinely row-varying. Values are five-seed, 95% dataset-bootstrap; the $\leq$8K and $\leq$20K populations agree.

Table 34: Centering analysis on each model's low-performing datasets ($\leq$ 20K population; 95% dataset-bootstrap intervals; higher macro-F1 is better). The class-wide part $(B - A)$ is negligible and slightly negative on two of three models; the test-re-centered row-varying correction $(D - A)$ recovers the full gain and the full AUC.

model	n	interval ΔF1 $(C-A)$	class-wide $(B-A)$	row-varying $(D-A)$	AUC gain $(C=D)$
TabPFN	81	$+2.26\,[+0.60, +4.34]$	$-0.92\,[-1.84, -0.21]$	$+2.27\,[+0.60, +4.37]$	$+1.95\,[+0.36, +3.92]$
TabICL	80	$+3.14\,[+0.61, +6.33]$	$+0.29\,[-0.29, +1.04]$	$+2.97\,[+0.45, +6.20]$	$+3.43\,[+0.66, +7.10]$
TabFM	79	$+4.89\,[+2.22, +7.97]$	$-0.80\,[-1.62, -0.14]$	$+4.92\,[+2.28, +7.99]$	$+4.10\,[+2.03, +6.50]$

The two component gains show no consistent correlation across datasets. Per-dataset Spearman correlations between the class-selection macro-F1 gain and the row-ordering ranking-AUC gain run from -0.26 to $+0.09$ across the five families with cached scores, with a pooled partial correlation of -0.03 given native validation macro-F1 (n=392); this does not rest on the by-construction row-side effect of the output-level head (Section 4.4).

A near-zero correlation is consistent with, but does not establish, distinct sources. TabPFN's raw coefficient is negative, and on most datasets the row correction does not deploy.

H.2 Second-stage learners and feature representations

The row-dependent gain is robust to the tested feature representations and second-stage learners; an alternative based on in-context prediction does not improve over the class-prior shift. Two stress tests follow, a factorial over feature representations and second-stage learners and an in-context-native alternative that repairs class selection at the input, both supporting the component as a property of the frozen model's predictions rather than an artifact of one fitted element.

Feature representation and learner choice. In a controlled study (226 low-performing model–dataset pairs, five seeds, three backbones) we measured the incremental macro-F1 gain beyond the class-prior shift for a factorial of feature representations $\times$ second-stage learners, at matched validation budget and parameter count. The quantile-bin representation of the main text is *not* distinguished: a raw-feature MLP recovers at least as much gain, permuted random bins tie the value-range bins, generic residual stackers (XGBoost, CatBoost, a random forest, ridge on probability residuals) all tie or slightly exceed every binned learner (best binned $-$ XGBoost $= -0.24\,[-0.83, +0.34]$), raising the discretization to 128–512 bins with equal-frequency or supervised (C4.5 recursive-entropy) boundaries leaves the gain flat in resolution, and supervised binning does not beat raw ($+0.19\,[-0.27, +0.69]$). Any competent second-stage learner recovers the row-ordering information, whatever the representation.

Context composition (an in-context-native alternative). We also tested repairing class selection at the input, resampling the demonstration set toward balanced class proportions ($q_c \propto \pi_c^{\alpha}$). Minority prediction responds monotonically to α, but a validation-selected composition beats trivial uniform balancing on no backbone, only $\sim$40–50% of the induced score change is class-wide, and it degrades accuracy, log-loss and Brier score comparably to the one-parameter shift while costing full re-inference.

These are validation-selected comparisons on one split, so they bound sensitivity to the second-stage learner rather than identify an optimal one.

H.3 Transfer of row corrections across model families

The correction direction is partly shared by trees and the from-scratch network, and a correction fitted on one model recovers 50–81% of another's own row-ordering gain. In *function* space (each model's centered correction evaluated on the same test rows against random-direction, label-permuted and row-permuted nulls; five seeds, 95% dataset bootstrap), in-context corrections align with one another well above the nulls (excess cosine $+0.74\,[+0.63, +0.84]$), but so does the from-scratch network ($+0.78\,[+0.69, +0.86]$), and tree pairs share a significant fraction (in-context–tree $+0.42\,[+0.23, +0.60]$; tree–tree $+0.53\,[+0.32, +0.70]$). Transfer tells the same story: a correction fitted on a source model recovers 50–81% of a target's own row-ordering gain, and averaging two in-context corrections for the held-out third yields $+2.42\,[+0.18, +5.21]$.

The shared direction is consistent with a property of weak tabular predictors rather than of in-context models specifically, and the probe analysis of Appendix I does not localize whether or where the signal is lost.

I Probing internal states for row-specific signal

The internal-state probes cannot localize the row-specific signal: their positive control fails, making the encoded-state null uninterpretable. To give the probes more rows, we re-extracted TabPFN and TabICL states for the training rows by 3-fold cross-fitting: each fold's rows pass through the model under a context built from the other folds, with native, validation and test posteriors on that same context, and hooked probabilities matched each model's own predictions on every dataset. The population is the 68 TabPFN and 58 TabICL datasets of that extraction (126 units), each probe fit on the held-out fold's rows (n_{fit} median 322, range 44–1,470) with λ and the deploy gate selected on the full validation set and a single test scoring; readouts pool the two backbones with backbone fixed effects (dataset-clustered bootstrap, $B = 4000$). The arms are binned features; raw values (the model's own standardized feature-value matrix); RF-cell, a fixed random per-cell ReLU encoding of those values; the query row's input-stage state (IN); the complete encoded token state (FULL); and IN and FULL from a weight-permuted encoder. ρ_X is arm X's gain divided by the binned-feature gain, and the decision rules were logged before the run.

Positive-control results. RF-cell recovers $\rho_{\text{RF-cell}} = 0.15\,[0.00, 0.51]$ gated ($0.15\,[-0.05, 0.60]$ ungated), and, gated, 0.11, 0.06 and -0.01 when the fit rows are subsampled to 60, 120 and 240 (Table 35). The binned features exceed the encoded states, but RF-cell recovers similarly little and the input state predicts the teacher correction as well as RF-cell does, so the registered outcome is *underpowered* under both standardizations, for first-member and ensemble-mean input states, and for gated and ungated gains. Boosted probes do not rescue the control ($\rho_{\text{RF-cell}} = 0.08$).

Table 35: Cross-fitted probes: macro-F1 gain and recovered fraction ρ relative to binned features (68 TabPFN + 58 TabICL datasets; backbone fixed effects; dataset-clustered 95% intervals; shared standardization; IN is the first-member input state). For IN, FULL and the permuted arms, ρ is the point estimate with its upper 95% bound in parentheses. The positive control (RF-cell) fails, so no row localizes whether or where the signal is lost.

probe input	gain (gated)	ρ (gated)	gain (ungated)	ρ (ungated)
binned features	$+3.50\,[+1.25, +6.18]$	1	$+3.34\,[+1.06, +6.03]$	1
raw values	$+0.25\,[-0.04, +0.51]$	–	$+0.27\,[-0.09, +0.62]$	–
RF-cell (control)	$+0.52\,[+0.01, +1.02]$	$+0.15\,[+0.00, +0.51]$	$+0.50\,[-0.14, +1.17]$	$+0.15\,[-0.05, +0.60]$
IN	$+0.32\,[-0.19, +0.79]$	+0.09 (0.36)	$+0.59\,[-0.03, +1.23]$	+0.18 (0.66)
FULL	$+0.42\,[+0.06, +0.83]$	+0.12 (0.35)	$+0.63\,[+0.21, +1.09]$	+0.19 (0.57)
permuted IN	$+0.11\,[-0.29, +0.52]$	+0.03 (0.21)	$+0.21\,[-0.33, +0.78]$	+0.06 (0.34)
permuted FULL	$+0.21\,[-0.17, +0.64]$	+0.06 (0.24)	$+0.18\,[-0.31, +0.70]$	+0.05 (0.30)

The probe family is basis-sensitive. RF-cell has many more columns than the binned features (d median 1,824 vs. 151 for TabPFN), so its failure could be a ridge-dimension effect. Re-running the same 126 units with dimension-matched dense arms — random ReLU features with 8 or 4 units per column, and the encoded IN and FULL states PCA-reduced to at most the binned-feature width — all fail (gated, registered standardization): $\rho_{\text{RF-cell},8} = 0.15\,[0.04, 0.51]$, $\rho_{\text{RF-cell},4} = 0.09\,[-0.03, 0.31]$, $\rho_{\text{PCA-IN}} = 0.09\,[-0.02, 0.31]$ and $\rho_{\text{PCA-FULL}} = 0.07\,[-0.07, 0.30]$. In the same run, one-hot bins with *random* boundaries and the same per-column bin counts recover as much as quantile bins at every fit size: $\rho_{\text{RB-bins}} = 1.10\,[0.93, 1.43]$ (1.03, 1.27 and 1.23 at 60, 120 and 240 fit rows). The probe's success is thus a property of the one-hot per-column basis, not of bin placement. The descriptive half-validation layerwise probes, the nonlinear ridge and boosted probe controls, and three further analyses — injecting a trained projection of the raw features into TabICL's in-context stack, fine-tuning under a pairwise ranking loss, and decoding the feature ranges from the encoded states — the positive control fails at those fit sizes too, so none of them is informative about whether or where the signal is lost.

A dense encoded state cannot be probed for this signal with this family: the encoded-state null is uninterpretable and the experiment does not localize the signal.

J Training interventions on decision and ranking gains

Table 36: Two-way dissociation: paired change in each remaining component gain (dataset-clustered 95% intervals). Balanced fine-tuning: TabPFN, n=66, against the version-matched frozen baseline; output-level feature head: three backbones pooled, n=99.

intervention	class selection (F1)	row ordering (AUC)
class-balanced training	$-4.00\,[-6.32, -1.77]$	$+0.11\,[-0.55, +0.77]$
centered output-level feature head	$-0.59\,[-1.61, +0.35]$	$-3.55\,[-5.87, -1.70]$

Class-balanced training substantially reduces the remaining class-selection gain while leaving the row-dependent gain; an output-level feature head has the opposite pattern. Fine-tuning updates the pretrained parameters, yet the class-selection component persists after *standard* fine-tuning: changing the training objective reduces it, a larger optimization budget or deeper updating does not. We audit TabPFN (Rubachev et al., 2025) and replicate the key comparison on TabICL, reporting the *remaining gain*, the test macro-F1 still recoverable by the class-prior shift after fine-tuning (lower is better), over 71 datasets for TabPFN (76 for TabICL) and five seeds per condition.

Effect of training conditions (Table 37). Class-balanced fine-tuning substantially reduces the class-selection component, and balancing only the output layer produces a similar reduction. No TabPFN condition removes the row-ordering component: after full-model balanced fine-tuning it still recovers +2.55 macro-F1 and +2.32 ranking-AUC points, so even full-model adaptation leaves feature-dependent structure the simple row correction can exploit. No condition raises the *achievable* (post-shift) macro-F1 either: standard fine-tuning leaves native macro-F1 unchanged (44.95 vs. frozen 45.18) and the balanced conditions raise it only by closing part of the gap the shift would close (achievable $\approx$ 51.6 for frozen and standard full against $\approx$ 49.6–49.8 balanced). The TabPFN audit therefore probes adaptation that changes decision behavior without improving the achievable predictor; the TabICL replication covers the complementary case.

Paired contrasts behind the main-text summary. On the fixed frozen-low-performing TabPFN population (n=66), class-balanced fine-tuning cuts the remaining class-selection gain by $-4.12\,[-6.29, -2.00]$ macro-F1 with output-layer balancing alone, while standard fine-tuning leaves it unchanged ($+0.43\,[-1.22, +2.02]$), and the row-ordering gain does not change significantly under any condition (balanced full-model $-0.06\,[-2.46, +2.22]$; difference between the two changes $-3.91\,[-6.94, -0.87]$). On the n=71 fine-tuning-eligible population the four Holm-adjusted primary contrasts in remaining gain are balanced full minus standard full $-4.68\,[-6.61, -2.84]$, balanced output-layer minus standard output-layer $-4.26\,[-6.00, -2.61]$, aggressive minus standard full $-0.79\,[-1.31, -0.33]$ (all three $p < 0.01$),

Table 37: Fine-tuning intervention panels: class-selection and row-ordering gains *remaining* after each condition (dataset-clustered 95% intervals; lower means more was removed). *Top:* TabPFN on the fixed frozen-low-performing population (n=66) — class-balanced fine-tuning substantially reduces class selection, output-layer balancing produces a similar reduction, and no condition removes row ordering. *Bottom:* TabICL full-model conditions (n=72 frozen-low-performing datasets, five seeds per condition) — balanced fine-tuning removes most of class selection with no differential row-ordering effect, and row ordering stays positive throughout.

condition	class-selection	row-ordering (macro-F1)	row-ordering (AUC)
TabPFN (n=66)			
frozen	+5.57 [+3.73, +7.56]	+2.61 [+0.57, +5.06]	+2.18 [+0.28, +4.50]
standard, output-layer	+5.28 [+2.96, +7.53]	+3.29 [+0.83, +6.20]	+2.22 [+0.60, +4.26]
balanced, output-layer	+1.45 [+0.30, +2.70]	+1.71 [+0.45, +3.27]	+1.97 [+0.69, +3.53]
standard, full	+6.00 [+3.95, +8.31]	+2.73 [+0.73, +5.25]	+2.27 [+0.63, +4.35]
balanced, full	+1.60 [+0.57, +2.81]	+2.55 [+0.68, +4.82]	+2.32 [+0.91, +4.06]
TabICL (n=72, five seeds per condition)			
no update	+3.44 [+1.97, +4.92]	+3.63 [+0.72, +7.38]	+3.80 [+0.89, +7.66]
standard, full	+2.08 [+1.06, +3.21]	+1.28 [−0.02, +3.08]	+1.62 [+0.38, +3.26]
balanced, full	+0.88 [−0.24, +2.03]	+1.45 [+0.22, +3.08]	+1.66 [+0.43, +3.27]

and standard output-layer minus standard full −0.20 [−0.94, +0.47]; a negative contrast favours the first-named condition.

Interaction test details. The selectivity interaction of Section 4.4 is $(\Delta_{\text{bal,class}} - \Delta_{\text{bal,row}}) - (\Delta_{\text{head,class}} - \Delta_{\text{head,row}})$, where Δ_{bal} and Δ_{head} are the changes under class-balanced fine-tuning and under the output-level feature head, "class" is the remaining class-prior shift macro-F1 gain and "row" the remaining ranking-AUC gain, each measured per dataset against the frozen baseline from the same pipeline. On TabPFN, balancing preferentially reduces class selection (−3.83 [−6.42, −1.23]) while the head preferentially reduces row ordering (−2.62 [−4.71, −0.88]), an interaction of −6.45 [−9.62, −3.30] (n=66); TabICL shows the same pattern (−3.02 [−5.17, −1.03], n=71) and pooling the two architectures gives −4.67 [−7.05, −2.42] (n=80). Because the class term is a macro-F1 gain and the row term a ranking-AUC gain, the interaction's magnitude is scale-dependent and should be read for sign only; the within-metric contrasts above carry the dissociation.

TabICL fine-tuning (Table 37). The dissociation replicates directionally. On 72 frozen-low-performing datasets with five fine-tuning seeds per condition, against the bit-exact no-update reference, balanced fine-tuning reduces class-selection error beyond standard fine-tuning (paired −1.20 [−2.03, −0.44] macro-F1) while the differential effect on row ordering is negligible (macro-F1 +0.16 [−0.28, +0.67]; ranking AUC +0.04 [−0.15, +0.24]), and row-ordering error remains positive after balanced fine-tuning (+1.45 [+0.22, +3.08] macro-F1, +1.66 [+0.43, +3.27] AUC). Output-layer-only conditions were not run for TabICL, so the depth result is specific to TabPFN, and standard full-model fine-tuning also reduces TabICL's row-ordering gain (−2.35 [−5.80, −0.09]), descriptively where fine-tuning materially improves the predictor.

Changing the objective closes the gap on both models. For TabPFN, class-balanced full-model fine-tuning reduces the remaining gain from +5.76 to +1.08 (paired −4.68 [−6.61, −2.84], Holm-adjusted), and updating only the 78K-parameter output layer suffices (+5.56 standard, +1.30 balanced), the tabular in-context analogue of long-tailed decoupling (Kang et al., 2020; Ren et al., 2020). On TabICL the remaining gain falls from +3.81 frozen and +2.60 after standard fine-tuning to +1.26 after balanced fine-tuning (paired −1.34 [−2.14, −0.61]), smaller but with the same signature (minority prediction rises to 0.64 of the observed rate, against 0.48 for standard). After balanced fine-tuning the model's own decisions already contain most of the repair, but they do not beat the one-parameter class-prior shift on achievable macro-F1: the model without updates plus the shift already reaches ≈ 51.6.

Balanced output-layer versus balanced full. The paired difference in remaining gain (balanced output-layer minus balanced full) is +0.22 [−0.21, +0.67] (n = 71, five seeds), so

its upper bound clears a one-point non-inferiority margin, chosen as small relative to the +5 to +6 standard-tuning gap it is weighed against. The result is margin-sensitive: a stricter 0.5-point margin would *not* be cleared, so we claim non-inferiority at the one-point level only, not equivalence. Uncorrected macro-F1 differs by $+0.00\,[-0.48, +0.37]$ and minority pred/obs by $+0.04\,[+0.02, +0.06]$.

Distinct effects of training interventions (Table 36). If the backbone leaves a feature signal unused, a feature head added to its output scores should remove the remaining row-ordering gain without reducing class selection; because the head skips the whole backbone, this tests the dissociation, not whether or where the signal is lost. We form $z_{\text{head}}(x) = z_{\text{native}}(x) + g(x)$, where g is a ridge head on the centered feature-range indicators fit on the training rows to the residuals of the cross-fitted *native* training scores, with per-class train-mean centering, so the head cannot express a class-wide shift and the backbone is untouched (a head cross-fitted on the training rows gives the same result), and measure both with the standard decomposition (predictions logged before results; not externally pre-registered). All three hold (Table 38): the remaining row-ordering gain collapses from $+3.57\,[+1.63, +5.86]$ to $+0.02\,[-0.25, +0.32]$ ranking-AUC (paired $-3.55\,[-5.87, -1.70]$; macro-F1 $-3.24\,[-5.30, -1.53]$), the pooled remaining class-selection gain is not significantly changed ($-0.59\,[-1.61, +0.35]$, still $+2.91\,[+1.45, +4.29]$ after the head), and the model with the head is natively better ($+3.79\,[+1.74, +6.18]$ ranking AUC, $+5.01\,[+2.73, +7.65]$ macro-F1). Mechanically the head is the ridge residual correction applied at the output, so its row-side effect is expected by construction; its class side is the empirical result, since centering removes only the training-set class-wide offset, and the class-side non-significance is a non-rejection, not an equivalence claim. The head is the intervention reciprocal to class balancing and localizes nothing.

Table 38: Output-level additive feature head, per backbone (low-performing populations under the recomputed gate, n=75/76/85; paired head-minus-frozen, dataset-clustered 95% intervals). The head removes the row-ordering residual; no backbone shows a significant class-selection change (TabICL borderline).

paired change	TabPFN	TabICL	TabFM
remaining row-ordering (AUC)	$-2.35\,[-4.33, -0.73]$	$-3.83\,[-7.47, -1.10]$	$-4.34\,[-6.69, -2.33]$
remaining row-ordering (F1)	$-1.44\,[-3.18, -0.17]$	$-3.20\,[-6.37, -0.73]$	$-4.87\,[-7.67, -2.40]$
remaining class-selection	$+0.31\,[-1.47, +2.04]$	$-1.29\,[-2.76, +0.03]$	$-0.76\,[-1.65, +0.10]$
native ranking AUC	$+2.54\,[+0.52, +4.91]$	$+4.23\,[+1.30, +8.03]$	$+4.50\,[+2.39, +6.97]$

Task-only head. To separate feature learning from residual fitting, we retrain g with the feature-only target of Section 4.3, for class c the quantity $\mathbf{1}[y{=}c] - \pi_c$ from the centered indicators, so the backbone's predictions are never observed during fitting and the training target is the only difference from the residual-trained head. All three hold (n=99 pooled): the remaining row-ordering AUC collapses (+3.57 to +0.23; change $-3.34\,[-5.61, -1.53]$), the pooled remaining class-selection gain is statistically unchanged ($-0.80\,[-1.93, +0.34]$, still $+2.70\,[+1.39, +4.15]$ after the head), and the native model improves ($+5.02\,[+2.71, +7.74]$ macro-F1). On TabICL alone the class-selection change is $-1.41\,[-2.69, -0.10]$, as with the residual-trained head.

Not every intervention could be run on every model family: the output-layer conditions exist only for TabPFN, TabFM has no fine-tuning interface, and the head is an output-level bypass. None of these experiments locates where the row-level signal is lost.

K Class-prior shifts and target-marginal adaptation

Changing the generative prior does not remove the class-prior shift gain: LDA gains +4.18 under its empirical prior and +1.89 under a uniform one, both excluding zero. Table 39 applies the same validation-tuned class-prior shift to conventional models fit on the same features, on TabPFN's low-performing datasets ($\leq$ 8K population, single seed except for the MLP): logistic regression and an MLP under ERM and class-balanced objectives, and Gaussian naive Bayes and LDA under their empirical and a uniform class prior, which

isolates the prior term while holding the class-conditional density fixed. Ranking AUC is unchanged by construction (Fact 1), so the shift corrects the decision threshold rather than substituting for target-marginal adaptation.

These are conventional models on TabPFN's low-performing datasets, so the comparison isolates the prior term within each learner; it does not rank learners.

Table 39: Marginal-adaptation control: class-prior shift macro-F1 gain for conventional models on TabPFN's low-performing datasets (higher is better; 95% dataset-clustered intervals). The in-context models leave +3.05 to +4.53 under the same shift.

model	training / prior	class-prior shift ΔF1	ΔAUC
logistic regression	ERM	+2.84 [+0.99, +4.71]	+0.00
logistic regression	class-balanced	+2.34 [+0.97, +3.94]	+0.00
MLP	ERM	+0.69 [−0.42, +1.82]	+0.00
MLP	class-balanced	+0.46 [−0.44, +1.35]	+0.00
Gaussian naive Bayes	empirical prior	+1.80 [+0.40, +3.32]	+0.00
Gaussian naive Bayes	uniform prior	+1.57 [+0.31, +2.99]	−0.00
LDA	empirical prior	+4.18 [+2.71, +5.72]	+0.00
LDA	uniform prior	+1.89 [+0.49, +3.33]	+0.00

K.1 Connections to thresholds, long tails, and stacking

Class-prior shifts, post-hoc threshold tuning, and feature-based score corrections each have precedents. Our comparisons quantify their distinct effects on class selection and row ranking in frozen tabular predictors. Tabular in-context models—TabPFN (Hollmann et al., 2023; 2025), TabICL (Qu et al., 2025) and TabFM (Kong and Das, 2026), with CARTE (Kim et al., 2024) for cross-table transfer—predict by conditioning on labelled rows in one forward pass, excelling on small data but trailing boosted trees more broadly (Grinsztajn et al., 2022; McElfresh et al., 2023; Erickson et al., 2025). TuneTables (Feuer et al., 2024) optimizes a learned context and Rubachev et al. (2025) study fine-tuning; we study the complementary low-performing, imbalanced regime. Biloš et al. (2026) analyse these models' internal readout mechanisms, which our probes lack the power to do (Appendix I), and McDowell et al. (2026) attribute thresholding's effectiveness for prior-data fitted networks to calibration, whereas Appendix E.5 shows the component surviving a recalibration that significantly improves class-wise ECE.

Our class-selection findings sit within long-tailed recognition, where imbalance-aware objectives (Lin et al., 2017; Cui et al., 2019; Cao et al., 2019; Ren et al., 2020), decoupled re-balancing (Kang et al., 2020) and label-distribution disentanglement (Hong et al., 2021) reshape training or post-adjust toward minority classes. Menon et al. (2021) formalize that argmax decisions under-select rare classes even under a correct posterior, and their post-hoc logit adjustment is precisely our class-wide correction (Section 3); that fix is by now shipped, `tabpfn` 9.0.0 exposing a per-class decision-threshold tuner (`ClassifierTuningConfig`) in the same functional family (Appendix F.4). The F-measure plug-in literature gives the corresponding account for macro-F1, where the optimal rule thresholds the posterior (Jansche, 2005; Dembczyński et al., 2011; Ye et al., 2012; Lipton et al., 2014) and plug-in rules are consistent for such non-decomposable measures, multiclass included (Koyejo et al., 2014; Narasimhan et al., 2014; Ramaswamy et al., 2015; Narasimhan et al., 2015; Dembczyński et al., 2017); Appendix E.3 confirms and extends this across families. On the tested datasets, unchanged in-context models show larger class-prior shift gains than the directly trained models (with split sensitivity for TabICL and TabFM). Much of this gain appears under labels drawn from each model's own posterior, while a feature-dependent correction provides a separate ranking gain.

Other relevant tools include calibration (Platt, 2000; Niculescu-Mizil and Caruana, 2005; Guo et al., 2017; Kull et al., 2019), discretized feature embeddings (Gorishniy et al., 2022; Fayyad and Irani, 1993; Dougherty et al., 1995), generalized additive models (Lou et al., 2012; Nori et al., 2019), stacking (Wolpert, 1992), logarithmic opinion pooling (Genest and Zidek, 1986;

Hinton, 2002), and the ROC invariance behind the ranking-preservation boundary (Elkan, 2001; Fawcett, 2006; Rahimi et al., 2020; Berta et al., 2024). Closest to our decomposition, Wang et al. (2026) split a long-tailed post-hoc *reranking* residual into a constant classwise and an input-dependent pairwise component, but their split orders candidate labels *within a row* and ours orders rows *within a class*. The decoupling of Kang et al. (2020) is orthogonal again: class-wise rescaling of a fixed classifier preserves row ordering whereas retraining its weights can reorder rows, so its two "classifier-only" interventions fall on opposite sides of our boundary.

These patterns occur across several model families; the class-selection magnitudes for TabICL and TabFM are sensitive to the train/test split (Appendix G).