A CROSS-VALIDATION GAIN DOES NOT IDENTIFY THE VALUE OF ITS SELECTION INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-validation changes two things at once. It gives the selector more validation information, and it usually changes the deployed predictor, because the fold-trained models are averaged. A gain from the complete pipeline cannot say which change produced it. We separate the two with an output-matched design that varies one component at a time, and add a third component that often travels with them: a small labelled panel that confirms the final choice. The evidence is a replay of four TabArena model families on 51 tasks, planned and sealed before any of the 765 held-back outer splits was opened. The value of selection information depends on what it is added to. For the validation-selected predictor, full out-of-fold selection raises ROC-AUC by 0.91 points when a single fold model is deployed but by only 0.28 points once fold outputs are averaged, while averaging alone is worth 0.98 points; both components lower normalized RMSE on all 13 regression tasks. An ordinary average of separately selected holdouts recovers about three quarters of the AUC gain of cross-validation. The same change of workflow matters far more for a benchmark than for deployment. RealMLP leads CatBoost on most splits of 16 tasks under a single holdout and of 32 tasks under cross-validation with averaging, yet validation picks RealMLP in 86 to 88 percent of splits either way. A confirmation panel of 32 labels that re-ranks all 800 saved models lowers AUC by 1.24 points on every binary task, and a shortlist removes most of that damage. A controlled study with actual refitting shows that the selection advantage survives full-data refitting and puts a cost on each policy.

1 INTRODUCTION

Validation results serve two different purposes. Benchmark designers use them to judge which model families look competitive under a given evaluation workflow, and practitioners use them to decide which validation-selected predictor to deploy. Cross-validation (CV) is usually credited with helping both, but a CV pipeline differs from a holdout pipeline in two ways. Its selector sees out-of-fold scores for every record, and its output is typically an average of the models trained on each fold. Better results from the complete pipeline do not say which of the two did the work, whether the chosen model family changed, or whether the extra training was worth its cost.

One pair of model families shows what is at stake. On the 765 held-back TabArena splits we study, moving from a single holdout to cross-validation with averaged fold models takes RealMLP from leading CatBoost on most splits of 16 tasks to leading on most splits of 32, while the family that validation would actually deploy changes in only one split in eight (Section 5). Which of the two changes deserves the credit, and which of the two results matters, depends on who is asking.

In current tabular benchmarks the two changes usually come together. TabArena (Erickson et al., 2025) evaluates each family through bagged, validation-selected ensembles and already notes the role of fold-model averaging. Tabular foundation models (Hollmann et al., 2025; Qu et al., 2025) and ensembled networks (Gorishniy et al., 2025) have made such comparisons more contested and more consequential. We ask a narrower question: how much of a comparison changes when selection information is added while the output construction is held fixed, and how much changes when predictions are averaged while the selector is held fixed?

We answer it with an output-matched design that crosses the two components (Figure 1) and evaluates the predictor the user would actually obtain. We then add a third component that often accompanies

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the other two, a small set of extra labels used to confirm the final choice. The main study replays saved predictions of four TabArena families on 51 tasks. All analysis choices were developed on one outer split and then frozen before any of the other 765 outer splits was opened. A second, local study refits the selected models on the full development pool, which the saved-prediction replay cannot do, and records the cost of every complete policy.

What we find:

- Averaging and selection information both improve the validation-selected predictor on held-back splits. They overlap heavily: on binary tasks each alone gives most of the gain of both, so the value credited to selection information depends on whether the outputs are averaged (Section 4).
- The benchmark comparison is far more sensitive to the workflow than the deployment decision is. The ranking of RealMLP against CatBoost reverses on many tasks, while the family chosen by validation barely changes (Section 5).
- The averaging gain tracks how much the fold-trained models disagree, as the ambiguity decomposition of Krogh & Vedelsby (1994) predicts (Section 6).
- A small confirmation panel that re-ranks every saved model damages a strong output on every task; restricting it to a shortlist repairs most of the damage but does not beat the native choice (Section 7).
- With actual refitting, a CV selection advantage survives, but ordinary refitted holdouts and equal-fit averages remain close competitors once cost is counted (Section 8).

2 Crossing selection with output construction

Let $f_{j,b}$ be configuration j fitted without validation fold b, for $b = 1, \ldots, B$. A selector returns nonnegative configuration weights w that sum to one, computed either from one held-out fold H_b or from the full out-of-fold matrix V. In both studies the selector is greedy ensemble selection (GES; Caruana et al., 2004), as used in auto-sklearn and AutoGluon (Feurer et al., 2022; Erickson et al., 2020). There are two levels of averaging. GES combines configurations; a fold-averaged output additionally combines the B fold-trained instances of each configuration:

$$p_{w,b}(x) = \sum_j w_j f_{j,b}(x), \qquad p_{w,\mathrm{avg}}(x) = \frac{1}{B}\sum_{b=1}^{B} p_{w,b}(x). \tag{1}$$

Crossing the selector with the output gives four cells (Figure 1). Moving along a row changes only the output construction; moving down a column changes only the selection information. For task t and a quality measure Q_t where larger is better, the two primary contrasts are

$$A_t = Q_t(H, \mathrm{avg}) - Q_t(H, \mathrm{single}), \qquad I_t = Q_t(V, \mathrm{avg}) - Q_t(H, \mathrm{avg}). \tag{2}$$

A_t adds averaging under one-fold selection and I_t adds selection information under averaged outputs. Their sum is the complete-pipeline difference along this path. The other path can allocate an interaction differently, so we also report the fourth cell and both reverse contrasts. An ordinary comparator, the *separate-fold average*, runs a separate holdout selector on each fold and averages the eight resulting outputs. It uses the same fitted models as CV but never pools selection scores, so it separates the benefit of pooled selection from the benefit of simply having every fold trained.

Two outcomes. For deployment, the practitioner's question, the family is chosen from validation scores alone and Q_t is the native test quality of the chosen predictor: ROC-AUC on binary tasks, log loss on multiclass tasks and RMSE on regression tasks. We divide RMSE by the standard deviation of the out-of-fold labels, fixed before any test lookup, and call the result NRMSE. The three metric strata are never pooled. For the benchmark question, Q_t is the mean of the four neural-versus-tree pairwise win indicators on task t, with ties worth one half. This unit puts all tasks on one scale but discards error magnitude, which is why we report both outcomes.

3 Studies and evidence stages

TabArena replay. We use all 51 TabArena tasks (Erickson et al., 2025) with public predictions: 30 binary, 8 multiclass and 13 regression. The primary families are CatBoost (Prokhorenkova et al.,

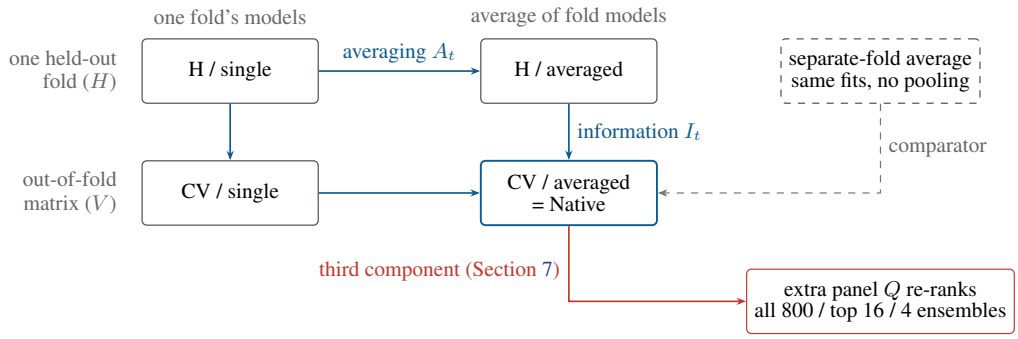

Figure 1. The components we vary. Within a row the configuration weights are identical; within a column both selectors feed the same output construction. "Single" is one fold-trained model bank, not necessarily one configuration. The two primary contrasts move along one axis at a time; both reverse paths are also reported. The separate-fold average uses the same fitted models as CV without pooling selection scores. A confirmation panel acts on the CV/averaged choice.

2018), RandomForest, RealMLP (Holzmüller et al., 2024) and NeuralNetTorch, each with 200 random configurations and eight saved fold-trained children per configuration. GES is a replay of the 40-round AutoGluon 1.3.1 implementation, including its tie handling and best-prefix rule. No model is trained; every quantity is computed from saved predictions.

Development and confirmation. All designs, estimands and the interpretation of the ordinary comparator were developed on outer split 0_0. The confirmatory study then covers every other available outer split, 765 task-splits in total (8 to 29 per task). Its plan, code and input hashes were sealed on 24 September 2026 before any new-split prediction was opened. Within every task-split, validation-only family choices were sealed first, then choices made with the confirmation panel, and only then were test outcomes scored; unit outcomes were not aggregated before the whole panel was complete. New splits reuse the same datasets and can overlap rows with the development split, so they test stability across resampling, not transfer to new tasks. Two further families, LightGBM (Ke et al., 2017) and TabM (Gorishniy et al., 2025), were fixed before their metadata was fetched and are reported as a separately adjusted six-family extension. The replay took 19.5 active hours on two CPU workers.

Inference. The statistical unit is the task. Outer splits are averaged within a task and tasks are weighted equally; neither splits nor family pairs count as independent units. Each prespecified family (deployment, six tests; benchmark, two; confirmation panel, twelve) uses Bonferroni-adjusted simultaneous 95% task-level t intervals and Holm-adjusted p values. Following the plan, a contrast counts as resolved only if its simultaneous interval excludes zero. Task bootstraps, leave-one-task-out means and sign counts are sensitivity checks (Appendix A); an independent re-implementation reproduced all 96 aggregate comparisons.

Local refit study. The TabArena archive contains no full-data refits, so refitting is studied locally on 18 OpenML classification tasks (Vanschoren et al., 2013) with 138 paired draws. Each development pool of about 5,120 records is split into three group-disjoint folds, and both selectors use the same 82 scikit-learn configurations (Pedregosa et al., 2011), preprocessing fitted inside each fold, and 50-round GES. Selected configurations are refitted on the complete pool with their weights unchanged. The outcome is utility $U = 100(1-L)$ for half-Brier loss L, so one point is 0.01 loss. The study performed 33,948 fold fits and 2,877 refits; it reuses development tasks and is not a held-out replication.

4 Both components improve the chosen predictor and move the benchmark

Figure 2 gives the main result and Table 1 the absolute levels. On binary tasks, averaging the fold-trained outputs of a fixed one-fold selection raised the AUC of the chosen predictor by 0.98 [0.33, 1.63] points, and all 30 tasks improved. Adding full out-of-fold selection to the averaged

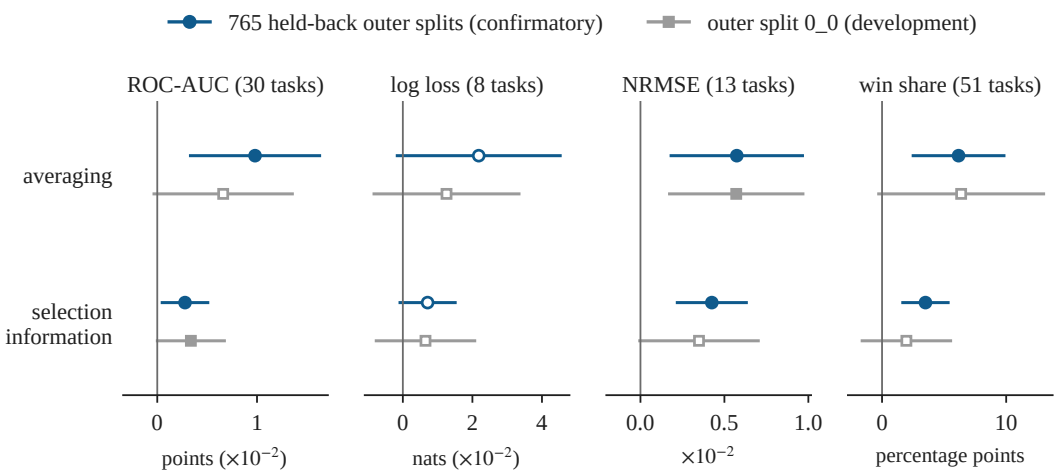

Figure 2. The two primary contrasts on the 765 held-back outer splits (blue circles) and on the development split (grey squares). The first three panels are the deployment outcome, the native quality of the validation-selected predictor, with positive values meaning better quality; the fourth is the benchmark outcome. Intervals are simultaneous 95% task-level intervals within each family. Filled markers: interval excludes zero.

Table 1. Absolute quality of the validation-selected predictor under each complete policy, averaged over outer splits within task and then over tasks (primary four families, 765 held-back splits).

Policy	ROC-AUC ↑	log loss ↓	NRMSE ↓	win share (%)
H / single	0.8381	0.2819	0.4120	53.7
H / averaged	0.8479	0.2601	0.4062	59.9
CV / single	0.8472	0.2662	0.4084	60.7
CV / averaged	0.8506	0.2529	0.4020	63.4
separate-fold average	0.8475	0.2546	0.4017	62.9

output added 0.28 [0.05, 0.51] points, with 26 of 30 tasks improving. On regression both components lowered NRMSE on every one of the 13 tasks, by 0.0057 [0.0018, 0.0096] for averaging and 0.0043 [0.0022, 0.0063] for information. On the eight multiclass tasks both effects point the same way in every task, and their Holm-adjusted p values are 0.023, but the simultaneous intervals include zero, so under our rule they are not resolved.

For every primary contrast, all leave-one-task-out means keep the sign of the full estimate, and the task-bootstrap intervals exclude zero in every stratum, the two multiclass contrasts included. The effects are skewed across tasks, however: the median task gains 0.32 AUC points from averaging against a mean of 0.98, so a minority of tasks accounts for much of the averaging gain even though every binary task gains something.

The reported model comparison moves as well. Averaging raised the neural-versus-tree win share by 6.2 [2.5, 9.8] percentage points and selection information by a further 3.5 [1.7, 5.3]. On the development split both win-share intervals had included zero. The held-back splits keep the averaging estimate almost unchanged (6.4 on the development split) and sharpen it. The information estimate is larger than on the development split (2.0), but its interval there was wide enough to contain it. Before the new splits were opened, a planning calculation put the 80%-power detectable information effect in AUC at 0.44 points if averaging over splits left the task-level variance unchanged, and 0.22 points if it halved it; the observed 0.28 was resolved.

The two components are partial substitutes (Figure 3). Selection information adds 0.91 AUC points when the output is a single fold's model but only 0.28 points once outputs are averaged. Averaging adds 0.98 points under one-fold selection but 0.35 under CV selection. The interaction, -0.63 points, is negative in 23 of 30 tasks, although its simultaneous interval $[-1.33, +0.07]$ just includes zero. The same pattern holds for win share. Any single-path decomposition therefore overstates whichever component is added first, and we report both paths for that reason.

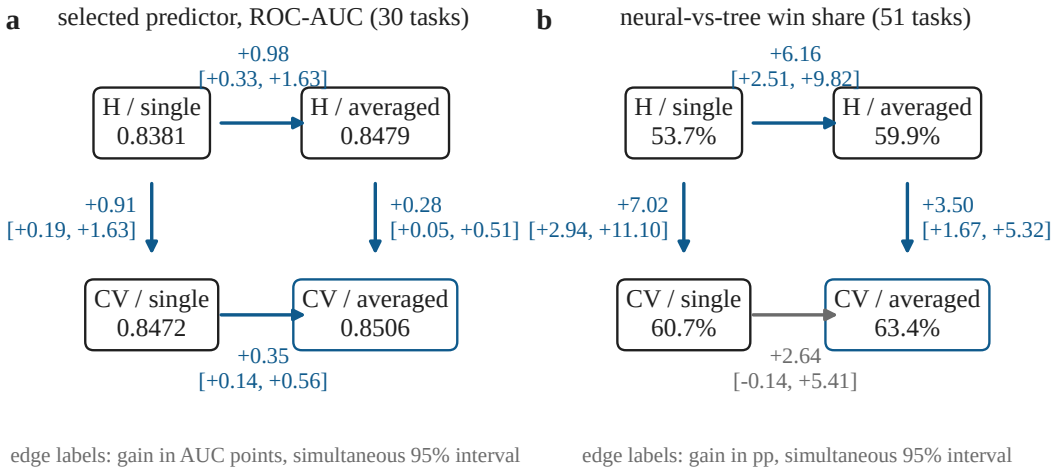

Figure 3. Both paths through the design. Corners are policy levels; edges are the gain from moving along them, with simultaneous 95% intervals. Blue edges exclude zero. Averaging is worth more when selection uses one fold, and selection information is worth more when the output is a single fold's model.

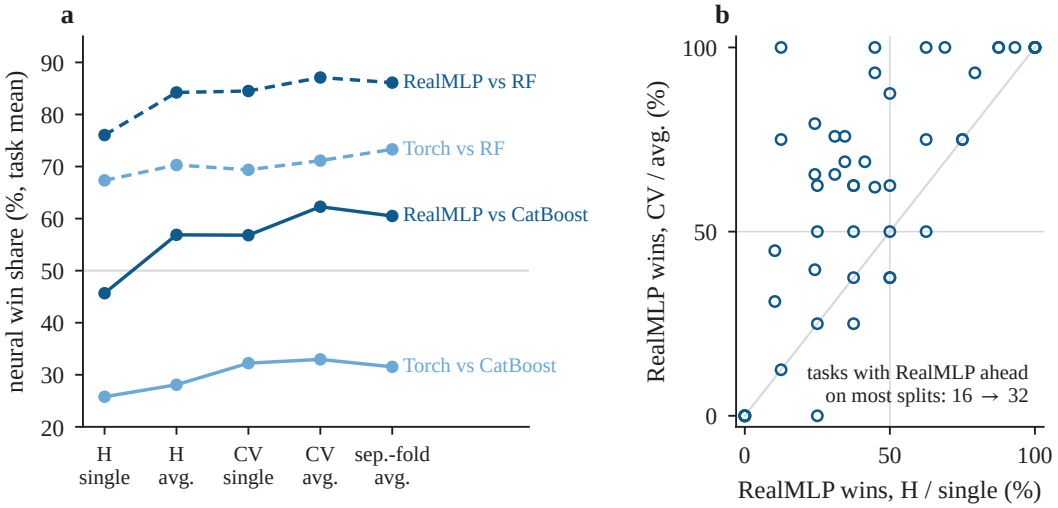

Figure 4. The neural-versus-tree comparison under each workflow, 765 held-back splits. (a) Share of splits on which the neural family has the lower native error, averaged over tasks, for the four pairs. (b) RealMLP against CatBoost, one point per task: the share of that task's splits on which RealMLP wins under H/single and under CV/averaged. Descriptive; the tests are those of Figure 2.

The ordinary separate-fold average comes close to full CV. Its AUC of 0.8475 recovers about three quarters of the gain from H/single to CV/averaged, and full CV is ahead by 0.31 [-0.004, 0.63] points (Holm $p = 0.038$). On regression the separate-fold average is as good as CV, with a difference of -0.0003 in NRMSE, and the win-share difference is 0.5 points. Most of what a CV pipeline adds over a single holdout is thus available from any procedure that trains and averages all fold models; pooled out-of-fold selection adds a smaller, binary-task-specific increment on top.

The six-family extension gives nearly identical deployment effects, because the two added families are rarely chosen. Its win-share effects are 5.7 [2.2, 9.3] points for averaging and 2.2 [0.5, 3.8] for information (Appendix A).

5 The comparison moves more than the choice

Figure 4 breaks the win-share effect into the four neural-versus-tree pairs. All four move in the same direction but by different amounts. The largest shift is for RealMLP against CatBoost, whose

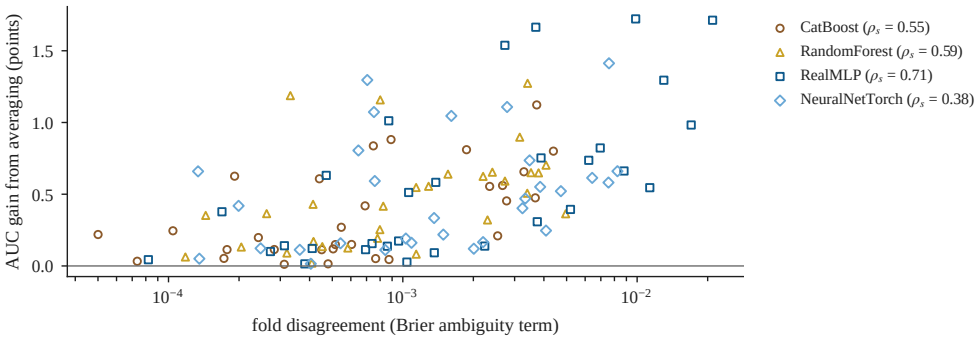

Figure 5. AUC gain from averaging against fold disagreement, one point per binary task and family (primary four families, one-fold selection weights, splits averaged within task). ρ_s is the Spearman correlation across the 30 tasks. Exploratory.

task-mean win share rises from 46% under H/single to 62% under CV/averaged, and the number of tasks on which RealMLP is ahead on most splits doubles, from 16 to 32. NeuralNetTorch against CatBoost moves from 26% to 33% and stays well below one half. Of the 204 task–pair comparisons, 34 change their majority winner between the two corners of the design (Table A.8). Someone reading a benchmark table would reach a different conclusion about RealMLP and CatBoost depending on which workflow produced it.

The family chosen for deployment, by contrast, changes little. Validation picks RealMLP in 661 of 765 task-splits under H/single and in 676 under CV/averaged. The chosen family differs between the two workflows in 92 task-splits, about one in eight, although 28 of the 51 tasks see at least one such change. Averaging by itself never changes the chosen family, since the family is chosen from the same validation scores; it only changes the predictor that is deployed. Most of the deployment gain in Section 4 must therefore come from better predictors inside the family that would have been chosen anyway. The same change of workflow can therefore matter a great deal for a benchmark table and very little for the model a practitioner ends up deploying.

6 Why averaging helps: fold disagreement

For half-Brier loss $L(p, y) = \frac{1}{2}\|p - y\|^2$, the average loss of the B fold models exceeds the loss of their mean by exactly their ambiguity, $\frac{1}{2B}\sum_b \|p_b - \bar{p}\|^2$ (Krogh & Vedelsby, 1994). The identity is exact but concerns Brier loss, not AUC. We therefore checked whether the AUC gain from averaging tracks ambiguity, computed for each binary task and family from the eight reconstructed fold predictors with the one-fold selection weights held fixed. The identity held to numerical precision in every case. Across tasks, the AUC gain from averaging rises with fold disagreement in all six families, with Spearman correlations from 0.38 for NeuralNetTorch to 0.71 for RealMLP (Figure 5; Appendix A). The pair that moves most in Section 5 is also the pair where disagreement differs most: RealMLP's fold models disagree more than CatBoost's on all 30 binary tasks, by a median factor of 2.2, and RealMLP gains more from averaging on 26 of them. NeuralNetTorch disagrees about as much as RealMLP but starts so far behind CatBoost that similar gains rarely change a winner. This is an exploratory correlation, not a causal claim. It does suggest how the averaging component can reorder families: on the tasks where a family's fold models disagree most, that family gains most from averaging, whatever the quality of its individual fold models.

7 A third component: confirming the final choice

The same question can be asked of a third component. Many pipelines set aside a small number of extra labels to confirm or revise the final choice. We tested this on the same held-back splits. From each split's test fold we drew a stratified panel Q of 32 or 128 labels, four times with fixed seeds, and scored every policy on the remaining records. Native is the CV/averaged, validation-selected predictor of Section 4. All-Q lets Q re-rank all 800 base predictions and the four family ensembles; Top16-Q lets it choose among Native and the 15 other predictors with the best out-of-fold scores; Four-Q

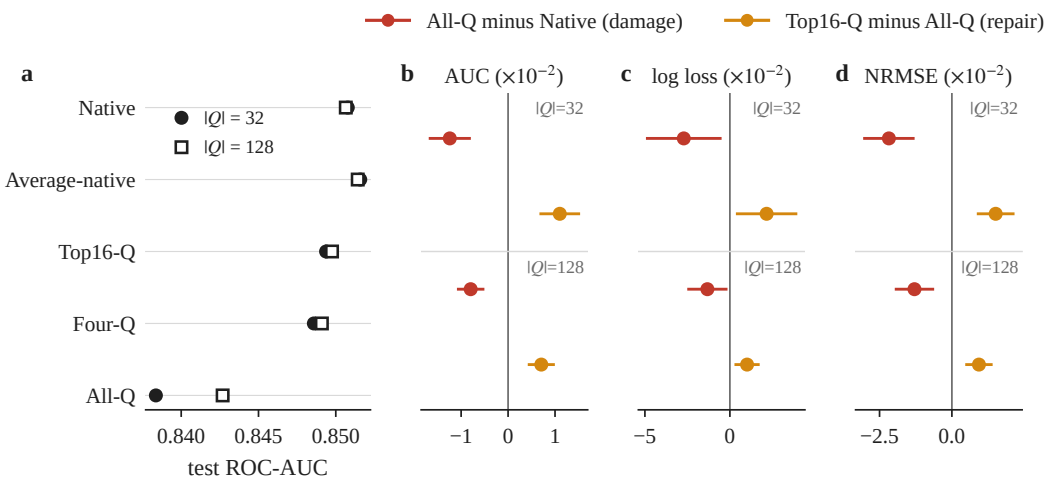

Figure 6. A confirmation panel on held-back splits. (a) Test ROC-AUC of each policy. (b–d) The twelve planned comparisons with simultaneous 95% intervals, in units of 10^{-2} of each native metric, positive meaning better. Every interval excludes zero.

chooses among the family ensembles; Average-native averages them without using Q. Because Q comes from the original test fold, its records are not an untouched test set, and we do not treat them as one.

Re-ranking everything was harmful on every task (Figure 6). With 32 labels, All-Q lowered AUC by 1.24 [0.79, 1.69] points and was worse than Native on all 30 binary tasks; it also raised log loss by 0.027 and NRMSE by 0.022, again on every task. With 128 labels the damage shrank to 0.80 AUC points, still on 29 of 30 binary tasks. Restricting the panel to the shortlist recovered 1.10 [0.67, 1.53] AUC points at $|Q| = 32$ and helped on every task in every stratum. Even so, Top16-Q stayed below Native (0.8494 against 0.8508), and simply averaging the four family ensembles without any extra labels was slightly better than both (0.8516; Appendix A). A bridge analysis that adds each family's default configuration reproduces all twelve comparisons.

The pattern is the one expected from selection among many candidates on few records (Ng, 1997; Bakshi et al., 2026): the panel is independent of the models, but its minimum is biased, and the bias grows with the number of candidates it may choose from. A strong system that already selects with thousands of out-of-fold records has little to gain from 32 or 128 more, and a lot to lose if those labels are allowed to overrule it. The six-family extension shows how large the loss can be. On one split of `concrete_compressive_strength`, All-Q twice chose a TabM configuration that scored best on its panel but whose archived predictions for two test records, absent from both panels, were wrong by five orders of magnitude. An audit traced these values to the original archive record rather than to our replay (Appendix B). Averaged over the task's splits, this one case widens the extension's regression intervals to include zero.

8 Refitting and cost in a controlled study

The replay evaluates fold-averaged outputs. Deployed systems often refit the chosen configurations on all data instead (auto-sklearn developers, 2026; AutoGluon developers, 2026), and that changes the predictor again. In the local study, the native CV fold average beat a single holdout by 0.229 utility points. With the CV weights fixed, averaging contributed 0.204 and selection on the same fold banks 0.025, and the ambiguity identity held to a residual of 2×10^{-14}. After both selectors received the same full-pool refit, CV kept an advantage of 0.045 [0.022, 0.069] points, positive on all 18 tasks, and the advantage grew by 0.020 [-0.001, 0.042] relative to the matched pre-refit comparison. This is a stable but small benefit: 0.00045 in half-Brier loss.

Figure 7 puts the policies on one scale of quality and work. A single holdout followed by refitting uses 39.4 CPU seconds per task-draw and reaches a loss of 0.1159; CV followed by refitting uses 106.6 seconds and reaches 0.1155. Ordinary three-holdout averaging, which uses the same 246 fits as

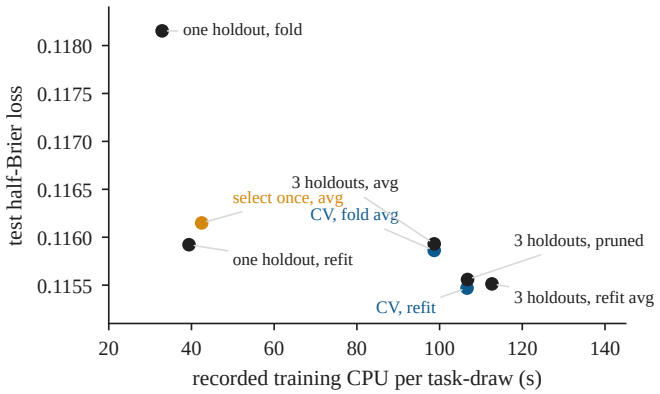

Figure 7. Quality and recorded training work of eight complete policies in the local study (equal-task means over 18 tasks and 138 draws). Lower and further left is better. Blue: cross-validation policies; orange: the select-once shortcut.

CV, leaves CV ahead by only 0.007 [-0.001, 0.015] points before refitting and 0.004 [-0.003, 0.012] after. A tempting shortcut, selecting on one holdout and then training only the chosen configurations' other fold models, needs 43% of CV's work but loses 0.029 [0.014, 0.044] points to CV on 17 of 18 tasks, and is both costlier and worse at the mean than a plain refitted holdout. Which component deserves credit and which policy to use are separate questions, and the second can only be answered by comparing complete alternatives together with their cost.

9 What an evaluation should report

The results suggest a few habits for anyone who compares tabular models through a validation pipeline or deploys the winner of such a comparison. A benchmark table should say how the evaluated predictor was built, not only how it was selected. "Cross-validated" can mean an out-of-fold selector with one refitted model, a fold ensemble, or both, and averaging alone moved RealMLP ahead of CatBoost on eleven more tasks (Table A.8). Because the two components overlap, the order in which they are added changes their apparent sizes by a factor of three; selection information is worth 0.91 AUC points with single outputs and 0.28 with averaged ones. Once the fold models exist, the full two-by-two table costs nothing to report. Claims about pooled out-of-fold selection should also be measured against the separate-fold average, which uses the same fits as CV and recovers three quarters of its binary-task gain and all of its regression gain.

For deployment, the quality of the predictor actually chosen belongs next to any win table. The majority winner of RealMLP against CatBoost changes on about a third of the tasks, while the chosen family changes on one split in eight. If a small confirmation panel is used at all, it should choose among a handful of candidates, and the native output should remain the reference against which it is judged.

10 Related work

Averaging, bagging and refitting. Averaging fold-trained models is a form of bagging (Breiman, 1996), and the ambiguity decomposition explains why disagreement helps (Krogh & Vedelsby, 1994). GES is our selector, not a contribution (Caruana et al., 2004); newer post-hoc ensemblers (Arango et al., 2025; Maier & Purucker, 2026) would plug into the same design. Auto-sklearn distinguishes fold-ensemble prediction from full-data refitting, and AutoGluon exposes refitting as a quality/cost choice (auto-sklearn developers, 2026; AutoGluon developers, 2026). Auto-sklearn 2.0 selects between holdout and CV per dataset (Feurer et al., 2022). We measure these familiar options side by side.

Cross-validation and model selection. The CV estimate targets the average error over training sets rather than the error of the fitted model (Bates et al., 2023). Selection on validation scores is

optimistic (Schneider et al., 2025; Schröder et al., 2025), reshuffling resampling splits can improve HPO (Nagler et al., 2024), and early stopping can make CV cheaper (Bergman et al., 2024). Our focus is different: which component of a CV pipeline improves the chosen predictor, and which changes a reported comparison.

Tabular benchmarks and evaluation noise. TabArena (Erickson et al., 2025) and TabRepo (Salinas & Erickson, 2024) made prediction-level replays like ours possible. TabArena already recognises the role of ensembling; our addition is the explicit crossing, the ordinary equal-fit comparator and the held-back confirmation. Reported rankings are fragile in several ways. Many published outperformance claims do not survive an honest account of evaluation noise (Christodoulou et al., 2025); single-holdout selection distorts comparisons on OpenML (Tschalzev et al., 2025); the fold assignment alone can reorder models (Hinduja et al., 2026); random rather than temporal splits change which methods look best (Rubachev et al., 2025); and selective reporting inflates leaderboard scores (Singh et al., 2025). Repeated CV reduces some of this variance (Eve et al., 2026). We add a less visible source: the construction of the evaluated predictor. The confirmation-panel result is an instance of the winner's curse in model selection (Ng, 1997; Bakshi et al., 2026). Active model selection chooses which examples to label so that few labels identify the best model (Okanovic et al., 2025); our panel is labelled at random and is meant as the simplest version of a common practice.

11 Limitations and conclusion

The replay covers four families (six in the extension), fixed random configuration menus and GES; it does not reproduce TabArena's full roster or leaderboard. The held-back splits reuse the development datasets, so they establish stability across resampling rather than transfer to new tasks, and the multiclass stratum has only eight tasks. The archive contains no full-data refits, so refitting is studied only in the smaller local study, whose tasks were used during development. No zero-containing interval establishes equivalence, and no minimum practically important difference was validated in advance.

Within these limits, the results suggest a simple procedure. When a pipeline changes both the selector and the output, each should be varied on its own while the final predictor is held fixed. In our data the two components overlap. Selection information is worth about three times as much without averaging as with it, and on binary tasks either component alone gives most of the gain of both. Both move the published comparison of families much more than they move the family a practitioner would deploy. A small confirmation panel is a third component, and it can undo the other two when it is allowed to choose among everything. Comparisons of model families should state the workflow under which they were made.

AI use statement

Generative AI assistants, including OpenAI Codex and Anthropic Claude, assisted with formulating research questions and hypotheses; developing the conceptual framework and mathematical and statistical expressions; designing and critiquing experiments; implementing and debugging data-processing, replay, refitting and analysis code; and interpreting results. They also assisted with translation and language editing, literature searches, manuscript drafting and revision, and the preparation of figures, tables and reproducibility artifacts. The benchmark labels and archived TabArena predictions were not synthesized by these assistants. Verification included input and selection hash checks, unit tests, and separate AI-assisted implementations that reconstructed numerical results from the saved artifacts. The authors are responsible for the final text, claims, references, code and other artifacts, including all AI-assisted contributions.

Ethics statement

This study is a secondary analysis of public benchmark datasets and archived predictions, with additional model fitting and refitting on public datasets. It involved no participant recruitment, intervention or new data collection from participants. Some source datasets contain human-derived records; public availability does not remove their privacy, bias or permitted-use considerations. The analysis compares evaluation workflows rather than identifying individuals, and the source datasets

and predictions remain subject to their original access conditions and licenses. The findings concern the evaluated tasks and workflows, not general rankings of model families or evidence of suitability for high-stakes deployment.

Reproducibility statement

The frozen plan, code, input hashes and per-member archive records (URL, ETag, byte range, CRC and SHA256) allow the replay to be rebuilt from the public TabArena archive; the 765 unit results, all sealed choices and the figure data will be released. The replay trained no models and made no paid calls; it transferred about 229 GB of public compressed predictions. An independent implementation reproduced all 96 aggregate comparisons, and a separate audit script reproduces the archive diagnosis of Appendix B offline from 19 archived record fragments. The local study's selected identities and refits were sealed before test scoring and an independent implementation reconstructed 2,622 prediction arrays (largest difference 4.4×10^{-16}).

References

Sebastian Pineda Arango, Maciej Janowski, Lennart Purucker, Arber Zela, Frank Hutter, and Josif Grabocka. Regularized neural ensemblers. In *Proceedings of the Fourth International Conference on Automated Machine Learning*, volume 293 of *PMLR*, pp. 8/1–33, 2025.

auto-sklearn developers. Resampling strategies: Holdout, cross-validation, and refitting. https://automl.github.io/auto-sklearn/master/examples/40_advanced/example_resampling.html, 2026.

AutoGluon developers. TabularPredictor.refit_full: API documentation. https://auto.gluon.ai/stable/api/autogluon.tabular.TabularPredictor.refit_full.html, 2026.

Soham Bakshi, Lingjun Gao, Zijun Gao, and Snigdha Panigrahi. Flexible inference for winners with conditional validity. arXiv:2607.18545, 2026.

Stephen Bates, Trevor Hastie, and Robert Tibshirani. Cross-validation: What does it estimate and how well does it do it? *Journal of the American Statistical Association*, 2023. doi: 10.1080/01621459.2023.2197686.

Edward Bergman, Lennart Purucker, and Frank Hutter. Don't waste your time: Early stopping cross-validation. In *Proceedings of the Third International Conference on Automated Machine Learning*, volume 256 of *PMLR*, 2024.

Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning*, 2004. doi: 10.1145/1015330.1015432.

Evangelia Christodoulou, Annika Reinke, Pascaline Andrè, Patrick Godau, Piotr Kalinowski, Rola Houhou, Selen Erkan, Carole H. Sudre, Ninon Burgos, Sofiène Boutaj, Sophie Loizillon, Maëlys Solal, Veronika Cheplygina, Charles Heitz, Michal Kozubek, Michela Antonelli, Nicola Rieke, Antoine Gilson, Leon D. Mayer, Minu D. Tizabi, M. Jorge Cardoso, Amber Simpson, Annette Kopp-Schneider, Gaël Varoquaux, Olivier Colliot, and Lena Maier-Hein. False promises in medical imaging AI? Assessing validity of outperformance claims. arXiv:2505.04720, 2025.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. AutoGluon-Tabular: Robust and accurate AutoML for structured data. arXiv:2003.06505, 2020.

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, Datasets and Benchmarks Track*, volume 38, 2025.

Célestin Eve, Gaël Varoquaux, and Thomas Moreau. Crossing the validation crisis: Cross-validation reduces benchmarking variance surprisingly well. arXiv:2606.12552, 2026.

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: Hands-free AutoML via meta-learning. *Journal of Machine Learning Research*, 23 (261):1–61, 2022.

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

Saurabh Hinduja, Gurmeet Kaur, Maneesh Bilalpur, Jeffrey Cohn, and Shaun Canavan. Beyond the fold: Quantifying split-level noise and the case for leave-one-dataset-out AU evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. arXiv:2604.02162.

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. doi: 10.1038/s41586-024-08328-6.

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

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, 1994.

Jannis Maier and Lennart Purucker. HAPEns: Hardware-aware post-hoc ensembling for tabular data. arXiv:2603.10582, 2026.

Thomas Nagler, Lennart Schneider, Bernd Bischl, and Matthias Feurer. Reshuffling resampling splits can improve generalization of hyperparameter optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Andrew Y. Ng. Preventing "overfitting" of cross-validation data. In *Proceedings of the 14th International Conference on Machine Learning*, 1997.

Patrik Okanovic, Andreas Kirsch, Jannes Kasper, Torsten Hoefler, Andreas Krause, and Nezihe Merve Gürel. All models are wrong, some are useful: Model selection with limited labels. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *PMLR*, pp. 2035–2043, 2025.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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*, volume 267 of *PMLR*, pp. 50817–50847, 2025.

Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, and Artem Babenko. TabReD: Analyzing pitfalls and filling the gaps in tabular deep learning benchmarks. In *International Conference on Learning Representations*, 2025.

David Salinas and Nick Erickson. TabRepo: A large scale repository of tabular model evaluations and its AutoML applications. In *Proceedings of the Third International Conference on Automated Machine Learning*, volume 256 of *PMLR*, 2024.

Lennart Schneider, Bernd Bischl, and Matthias Feurer. Overtuning in hyperparameter optimization. In *Proceedings of the Fourth International Conference on Automated Machine Learning*, volume 293 of *PMLR*, 2025.

Sietse Schröder, Mitra Baratchi, and Jan N. van Rijn. Overfitting in combined algorithm selection and hyperparameter optimization. In *Advances in Intelligent Data Analysis XXIII (IDA 2025)*, volume 15669 of *Lecture Notes in Computer Science*. Springer, 2025. doi: 10.1007/978-3-031-91398-3_14.

Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D'Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah A. Smith, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. The leaderboard illusion. arXiv:2504.20879, 2025.

Andrej Tschalzev, Lennart Purucker, Stefan Lüdtke, Frank Hutter, Christian Bartelt, and Heiner Stuckenschmidt. Unreflected use of tabular data repositories can undermine research quality. ICLR 2025 Workshop on the Future of Machine Learning Data Practices and Repositories; arXiv:2503.09159, 2025.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

A ALL PLANNED COMPARISON FAMILIES

Units are native: ROC-AUC on its $[0, 1]$ scale, log loss in nats, NRMSE as a fraction of the out-of-fold label standard deviation, and win share as a fraction. Positive values always mean better quality. "$+/-/0$" counts task means; "LOO range" is the range of leave-one-task-out means. Averaging (H) is A_t and information (avg.) is I_t; information (single) and averaging (CV) are the reverse path, and the interaction is information under averaged minus information under single outputs.

A.1 PRIMARY DEPLOYMENT AND BENCHMARK FAMILIES

Table A.1 lists the six deployment tests of Section 4. Four simultaneous intervals exclude zero: averaging and information in ROC-AUC and in NRMSE. The two multiclass contrasts point in the same direction on all eight tasks and have Holm $p = 0.023$, but their simultaneous intervals include zero, so under the prespecified rule they count as unresolved. Both benchmark tests in Table A.2 are resolved. In every row the leave-one-task-out range stays on the side of zero taken by the full estimate.

Table A.1. Primary deployment family (six tests).

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
averaging (H)	AUC	30	+0.0098	[+0.0033, +0.0163]	0.00091	30/0/0	[+0.0084, +0.0101]
information (avg.)	AUC	30	+0.0028	[+0.0005, +0.0051]	0.0053	26/4/0	[+0.0022, +0.0030]
averaging (H)	log loss	8	+0.0218	[−0.0016, +0.0452]	0.023	8/0/0	[+0.0180, +0.0248]
information (avg.)	log loss	8	+0.0071	[−0.0008, +0.0150]	0.023	8/0/0	[+0.0053, +0.0080]
averaging (H)	NRMSE	13	+0.0057	[+0.0018, +0.0096]	0.0023	13/0/0	[+0.0051, +0.0062]
information (avg.)	NRMSE	13	+0.0043	[+0.0022, +0.0063]	0.00017	13/0/0	[+0.0038, +0.0045]

Table A.2. Primary benchmark family (two tests).

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
averaging (H)	win share	51	+0.0616	[+0.0251, +0.0982]	0.00029	28/11/12	[+0.0567, +0.0666]
information (avg.)	win share	51	+0.0350	[+0.0167, +0.0532]	0.0001	30/8/13	[+0.0319, +0.0382]

A.2 Reverse path, interaction and the ordinary comparator

The supporting family in Table A.3 holds the two reverse-path contrasts, their interaction and the separate-fold comparator for each stratum. The reverse path makes the substitution of Figure 3 explicit. Selection information adds 0.0091 [0.0019, 0.0163] AUC when the output is a single fold's model, three times its value under averaged outputs, and averaging under CV selection adds 0.0035 [0.0014, 0.0056]. The AUC interaction is negative on 23 of 30 tasks, but its interval $[-0.0133, +0.0007]$ includes zero. Full CV leads the separate-fold average by 0.0031 AUC (Holm $p = 0.038$, simultaneous interval including zero), and the two are level on regression and on win share.

Table A.3. Prespecified supporting family: reverse path, interaction and the separate-fold comparator (sixteen tests).

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
CV vs separate-fold	AUC	30	+0.0031	[−0.0000, +0.0063]	0.038	23/7/0	[+0.0026, +0.0034]
information (single)	AUC	30	+0.0091	[+0.0019, +0.0163]	0.0043	29/1/0	[+0.0075, +0.0095]
averaging (CV)	AUC	30	+0.0035	[+0.0014, +0.0056]	0.00018	28/2/0	[+0.0032, +0.0037]
interaction	AUC	30	−0.0063	[−0.0133, +0.0007]	0.06	7/23/0	[−0.0068, −0.0047]
CV vs separate-fold	log loss	8	+0.0016	[−0.0019, +0.0052]	0.33	7/1/0	[+0.0011, +0.0022]
information (single)	log loss	8	+0.0157	[−0.0033, +0.0347]	0.066	8/0/0	[+0.0123, +0.0178]
averaging (CV)	log loss	8	+0.0132	[−0.0060, +0.0325]	0.12	8/0/0	[+0.0102, +0.0150]
interaction	log loss	8	−0.0086	[−0.0205, +0.0033]	0.11	1/7/0	[−0.0102, −0.0070]
CV vs separate-fold	NRMSE	13	−0.0003	[−0.0019, +0.0013]	1	5/8/0	[−0.0006, −0.0001]
information (single)	NRMSE	13	+0.0035	[+0.0003, +0.0068]	0.021	13/0/0	[+0.0030, +0.0038]
averaging (CV)	NRMSE	13	+0.0065	[+0.0026, +0.0103]	0.00065	13/0/0	[+0.0059, +0.0069]
interaction	NRMSE	13	+0.0007	[−0.0023, +0.0038]	1	9/4/0	[+0.0004, +0.0013]
CV vs separate-fold	win share	51	+0.0051	[−0.0189, +0.0291]	1	22/16/13	[+0.0028, +0.0083]
information (single)	win share	51	+0.0702	[+0.0294, +0.1110]	3.6e-05	34/7/10	[+0.0656, +0.0741]
averaging (CV)	win share	51	+0.0264	[−0.0014, +0.0541]	0.048	25/10/16	[+0.0238, +0.0288]
interaction	win share	51	−0.0353	[−0.0814, +0.0108]	0.12	14/27/10	[−0.0391, −0.0298]

A.3 Confirmation panel

All twelve panel tests in Table A.4 exclude zero, and the sign counts show that the damage from All-Q and the repair from the shortlist occur on nearly every task. The absolute levels in Table A.5 add a detail that the contrasts do not show. Averaging the four family ensembles without extra labels is slightly better than Native in AUC (0.8516 against 0.8508 at $|Q| = 32$) but worse in log loss and NRMSE, so it is not a uniform improvement. The bridge analysis in Table A.6, which adds each family's default configuration to the menu, reproduces every comparison.

Table A.4. Primary confirmation-panel family, 200-configuration menu (twelve tests).

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
Q32: All-Q − Native	AUC	30	−0.0124	[−0.0169, −0.0079]	2.1e-08	0/30/0	[−0.0128, −0.0119]
Q32: Top16-Q − All-Q	AUC	30	+0.0110	[+0.0067, +0.0153]	1e-07	30/0/0	[+0.0103, +0.0114]
Q128: All-Q − Native	AUC	30	−0.0080	[−0.0109, −0.0050]	2.7e-08	1/29/0	[−0.0083, −0.0076]
Q128: Top16-Q − All-Q	AUC	30	+0.0071	[+0.0042, +0.0100]	1.9e-07	30/0/0	[+0.0067, +0.0073]
Q32: All-Q − Native	log loss	8	−0.0271	[−0.0494, −0.0049]	0.0042	0/8/0	[−0.0294, −0.0229]
Q32: Top16-Q − All-Q	log loss	8	+0.0216	[+0.0035, +0.0397]	0.0042	8/0/0	[+0.0191, +0.0245]
Q128: All-Q − Native	log loss	8	−0.0132	[−0.0251, −0.0014]	0.0042	0/8/0	[−0.0143, −0.0114]
Q128: Top16-Q − All-Q	log loss	8	+0.0101	[+0.0027, +0.0175]	0.003	8/0/0	[+0.0086, +0.0109]
Q32: All-Q − Native	NRMSE	13	−0.0218	[−0.0307, −0.0129]	1.4e-05	0/13/0	[−0.0228, −0.0199]
Q32: Top16-Q − All-Q	NRMSE	13	+0.0151	[+0.0086, +0.0216]	2.1e-05	13/0/0	[+0.0142, +0.0159]
Q128: All-Q − Native	NRMSE	13	−0.0129	[−0.0197, −0.0061]	0.00011	0/13/0	[−0.0137, −0.0115]
Q128: Top16-Q − All-Q	NRMSE	13	+0.0093	[+0.0046, +0.0141]	9e-05	13/0/0	[+0.0086, +0.0099]

A.4 Fold disagreement and pairwise comparisons

The correlation between fold disagreement and the AUC gain from averaging is positive for every family under both weight vectors (Table A.7): Spearman correlations range from 0.38 to 0.71 with one-fold weights and from 0.44 to 0.63 with CV weights. The pooled rows are descriptive; the 180 task–family rows are not independent. Table A.8 gives the pairwise counts behind Figure 4 next to

Table A.5. Absolute levels in the confirmation-panel study. Log loss and NRMSE are shown as losses (lower is better).

Stratum	$\|Q\|$	Native	Average-native	Top16-Q	Four-Q	All-Q
AUC	32	0.8508	0.8516	0.8494	0.8486	0.8384
AUC	128	0.8506	0.8514	0.8498	0.8491	0.8427
log loss	32	0.2504	0.2536	0.2560	0.2556	0.2776
log loss	128	0.2541	0.2570	0.2572	0.2567	0.2673
NRMSE	32	0.4018	0.4065	0.4085	0.4093	0.4236
NRMSE	128	0.4019	0.4065	0.4055	0.4048	0.4148

Table A.6. Bridge analysis: the same panel family with each family's default configuration added (201-configuration menu).

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
Q32: All-Q − Native	AUC	30	−0.0126	[−0.0170, −0.0081]	1.4e-08	0/30/0	[−0.0130, −0.0120]
Q32: Top16-Q − All-Q	AUC	30	+0.0111	[+0.0068, +0.0154]	7.3e-08	30/0/0	[+0.0104, +0.0115]
Q128: All-Q − Native	AUC	30	−0.0080	[−0.0109, −0.0050]	3e-08	1/29/0	[−0.0083, −0.0076]
Q128: Top16-Q − All-Q	AUC	30	+0.0071	[+0.0042, +0.0100]	1.6e-07	30/0/0	[+0.0067, +0.0073]
Q32: All-Q − Native	log loss	8	−0.0273	[−0.0496, −0.0050]	0.0042	0/8/0	[−0.0296, −0.0231]
Q32: Top16-Q − All-Q	log loss	8	+0.0218	[+0.0037, +0.0399]	0.0042	8/0/0	[+0.0193, +0.0247]
Q128: All-Q − Native	log loss	8	−0.0133	[−0.0250, −0.0015]	0.0042	0/8/0	[−0.0143, −0.0114]
Q128: Top16-Q − All-Q	log loss	8	+0.0101	[+0.0027, +0.0176]	0.0031	8/0/0	[+0.0086, +0.0110]
Q32: All-Q − Native	NRMSE	13	−0.0218	[−0.0310, −0.0127]	1.7e-05	0/13/0	[−0.0229, −0.0199]
Q32: Top16-Q − All-Q	NRMSE	13	+0.0152	[+0.0088, +0.0217]	1.8e-05	13/0/0	[+0.0143, +0.0160]
Q128: All-Q − Native	NRMSE	13	−0.0129	[−0.0196, −0.0061]	0.00011	0/13/0	[−0.0137, −0.0114]
Q128: Top16-Q − All-Q	NRMSE	13	+0.0093	[+0.0047, +0.0140]	7.9e-05	13/0/0	[+0.0085, +0.0098]

the development-split counts from the original analysis. The development counts are single-split win totals and are therefore noisier; they agree in direction with the held-back majorities for every pair.

Table A.7. Fold disagreement and AUC gain from averaging: correlations across the 30 binary tasks, per family and weight vector. Exploratory; the pooled row is descriptive and does not have six times as many independent units.

Family	Weights	Rows	Pearson	Spearman
CatBoost	H0	30	0.61	0.55
RandomForest	H0	30	0.43	0.59
RealMLP	H0	30	0.65	0.71
NeuralNetTorch	H0	30	0.35	0.38
LightGBM	H0	30	0.47	0.44
TabM	H0	30	0.62	0.61
all six (descriptive)	H0	180	0.57	0.61
CatBoost	CV	30	0.54	0.51
RandomForest	CV	30	0.45	0.49
RealMLP	CV	30	0.53	0.63
NeuralNetTorch	CV	30	0.41	0.54
LightGBM	CV	30	0.47	0.44
TabM	CV	30	0.49	0.59
all six (descriptive)	CV	180	0.48	0.54

B SIX-FAMILY EXTENSION

LightGBM and TabM are added to the trees and the neural networks, respectively, giving nine neural-versus-tree pairs. The extension is adjusted separately and cannot rescue or replace a primary result. Its regression panel is dominated by one task: on `concrete_compressive_strength`, averaged over the task's 29 held-back splits, All-Q exceeded Native's NRMSE by 6.95 at $|Q| = 32$ and 8.35 at $|Q| = 128$. The leave-one-task-out range shows that without that task the regression damage is of the same size as in the primary family.

The extreme contribution comes from split 4_0 among the task's 29 held-back splits. In two of its eight panel draws, one at $|Q| = 32$ and one at $|Q| = 128$, All-Q selected `TabM_r3_BAG_L1`. Its

Table A.8. Neural-versus-tree pairs under each workflow. "Share" is the task mean of the share of held-back splits the neural family wins; "majority tasks" counts tasks where it wins most splits; the development split row gives the win counts over 51 tasks on split 0_0 from the original analysis.

Pair	Summary	H/single	H/avg.	CV/single	CV/avg.	sep.-fold
RealMLP vs CatBoost	held-back, share (%)	45.7	56.9	56.8	62.3	60.5
	held-back, majority tasks	16	27	28	32	33
	split 0_0, wins	25	34	30	36	33
RealMLP vs RF	held-back, share (%)	76.0	84.2	84.5	87.1	86.1
	held-back, majority tasks	39	45	46	47	47
	split 0_0, wins	39	44	43	45	45
Torch vs CatBoost	held-back, share (%)	25.8	28.1	32.2	33.0	31.5
	held-back, majority tasks	9	12	10	11	12
	split 0_0, wins	12	11	11	13	16
Torch vs RF	held-back, share (%)	67.3	70.3	69.4	71.1	73.3
	held-back, majority tasks	33	36	35	37	38
	split 0_0, wins	35	35	35	34	36

excess NRMSE over Native was 805.271 and 967.857, respectively. Both panels omitted the two extreme test records, on which the released fold-averaged predictions were approximately $-168{,}744$ and $-168{,}740$, against labels 6.267 and 12.638. The normalization uses the out-of-fold label standard deviation, 16.772, rather than a test-dependent scale.

A targeted audit reacquired the original ZIP member and verified its CRC and exact SHA-256 agreement with the replay input. Direct extraction of the serialized float arrays also reproduced the extreme values. The anomaly is already present in saved children `S1F7` and `S1F8`; their eight-child average reconstructs the released ensemble predictions to float32 precision. The source record itself reports test RMSE 12,867.1754, consistent with recomputation from the released predictions (12,867.1748; relative difference 4.63×10^{-8}), despite an out-of-fold RMSE of 3.8333. These checks locate the failure upstream of our transfer, decoding and replay, rather than in our analysis pipeline. Distinguishing fitting instability from an upstream preprocessing or inference defect would require the saved models or execution traces. The configuration and its outcomes remain in the frozen analysis.

Tables B.1–B.4 repeat the primary families with six families. The deployment effects are nearly unchanged, because LightGBM and TabM are rarely chosen by validation. With nine pairs instead of four, the win-share effect of selection information is smaller, 2.2 [0.5, 3.8] percentage points, while averaging keeps most of its effect.

Table B.1. Extension deployment family.

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
averaging (H)	AUC	30	+0.0098	[+0.0033, +0.0163]	0.00091	30/0/0	[+0.0084, +0.0101]
information (avg.)	AUC	30	+0.0028	[+0.0005, +0.0051]	0.0052	26/4/0	[+0.0022, +0.0030]
averaging (H)	log loss	8	+0.0218	[−0.0016, +0.0452]	0.023	8/0/0	[+0.0180, +0.0248]
information (avg.)	log loss	8	+0.0071	[−0.0008, +0.0150]	0.023	8/0/0	[+0.0053, +0.0080]
averaging (H)	NRMSE	13	+0.0056	[+0.0016, +0.0096]	0.0036	13/0/0	[+0.0050, +0.0061]
information (avg.)	NRMSE	13	+0.0043	[+0.0022, +0.0064]	0.0002	13/0/0	[+0.0039, +0.0046]

Table B.2. Extension benchmark family.

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
averaging (H)	win share	51	+0.0573	[+0.0220, +0.0926]	0.00093	34/12/5	[+0.0516, +0.0618]
information (avg.)	win share	51	+0.0215	[+0.0049, +0.0382]	0.0044	32/13/6	[+0.0186, +0.0235]

C LOCAL REFIT STUDY

The two frozen primary quantities of the local study were the CV-minus-holdout utility after common refitting, +0.0453 [+0.0217, +0.0689] (Holm $p = 0.0004$), and its change relative to the output-matched pre-refit comparison, +0.0202 [-0.0011, +0.0415] (Holm $p = 0.032$). The Holm p of the

Table B.3. Extension supporting family.

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
CV vs separate-fold	AUC	30	+0.0031	[−0.0000, +0.0063]	0.038	23/7/0	[+0.0026, +0.0034]
information (single)	AUC	30	+0.0091	[+0.0019, +0.0163]	0.0043	29/1/0	[+0.0075, +0.0095]
averaging (CV)	AUC	30	+0.0035	[+0.0014, +0.0056]	0.00019	28/2/0	[+0.0032, +0.0037]
interaction	AUC	30	−0.0063	[−0.0133, +0.0007]	0.067	7/23/0	[−0.0068, −0.0047]
CV vs separate-fold	log loss	8	+0.0016	[−0.0020, +0.0052]	0.35	6/2/0	[+0.0011, +0.0022]
information (single)	log loss	8	+0.0157	[−0.0034, +0.0348]	0.075	8/0/0	[+0.0123, +0.0178]
averaging (CV)	log loss	8	+0.0132	[−0.0060, +0.0325]	0.11	8/0/0	[+0.0102, +0.0150]
interaction	log loss	8	−0.0086	[−0.0205, +0.0034]	0.11	2/6/0	[−0.0102, −0.0070]
CV vs separate-fold	NRMSE	13	−0.0003	[−0.0020, +0.0013]	0.98	6/7/0	[−0.0006, −0.0001]
information (single)	NRMSE	13	+0.0035	[+0.0002, +0.0067]	0.023	13/0/0	[+0.0030, +0.0037]
averaging (CV)	NRMSE	13	+0.0064	[+0.0026, +0.0103]	0.00073	13/0/0	[+0.0059, +0.0069]
interaction	NRMSE	13	+0.0008	[−0.0022, +0.0039]	0.98	8/5/0	[+0.0005, +0.0015]
CV vs separate-fold	win share	51	+0.0040	[−0.0237, +0.0316]	0.98	21/23/7	[+0.0015, +0.0082]
information (single)	win share	51	+0.0550	[+0.0196, +0.0904]	0.0002	35/11/5	[+0.0514, +0.0592]
averaging (CV)	win share	51	+0.0238	[−0.0045, +0.0521]	0.095	28/15/8	[+0.0208, +0.0265]
interaction	win share	51	−0.0335	[−0.0760, +0.0090]	0.11	20/27/4	[−0.0372, −0.0282]

Table B.4. Extension confirmation-panel family, 200-configuration menu.

Contrast	Stratum	Tasks	Mean	Simultaneous 95%	Holm p	$+/-/0$	LOO range
Q32: All-Q − Native	AUC	30	−0.0122	[−0.0169, −0.0074]	9.9e-08	0/30/0	[−0.0126, −0.0115]
Q32: Top16-Q − All-Q	AUC	30	+0.0115	[+0.0069, +0.0161]	1.7e-07	30/0/0	[+0.0108, +0.0119]
Q128: All-Q − Native	AUC	30	−0.0075	[−0.0105, −0.0045]	1.7e-07	1/29/0	[−0.0078, −0.0070]
Q128: Top16-Q − All-Q	AUC	30	+0.0077	[+0.0047, +0.0107]	9.9e-08	30/0/0	[+0.0073, +0.0080]
Q32: All-Q − Native	log loss	8	−0.0277	[−0.0502, −0.0052]	0.0099	0/8/0	[−0.0300, −0.0232]
Q32: Top16-Q − All-Q	log loss	8	+0.0232	[+0.0046, +0.0417]	0.0099	8/0/0	[+0.0209, +0.0263]
Q128: All-Q − Native	log loss	8	−0.0124	[−0.0225, −0.0022]	0.0099	0/8/0	[−0.0133, −0.0109]
Q128: Top16-Q − All-Q	log loss	8	+0.0101	[+0.0013, +0.0189]	0.0099	8/0/0	[+0.0082, +0.0109]
Q32: All-Q − Native	NRMSE	13	−0.5546	[−2.4356, +1.3263]	1	0/13/0	[−0.6001, −0.0214]
Q32: Top16-Q − All-Q	NRMSE	13	+0.5491	[−1.3319, +2.4300]	1	13/0/0	[+0.0158, +0.5941]
Q128: All-Q − Native	NRMSE	13	−0.6547	[−2.9168, +1.6074]	1	0/13/0	[−0.7091, −0.0134]
Q128: Top16-Q − All-Q	NRMSE	13	+0.6513	[−1.6114, +2.9140]	1	13/0/0	[+0.0098, +0.7053]

Table C.1. Complete local policies: quality and recorded work, equal-task means over 18 tasks. Training CPU includes each policy's base fitting, validation prediction, selection and refitting, even when execution reused a cache; prediction CPU sums component work on the test batch and is not a serving-latency measurement. The pruned average and the select-once shortcut were added after the first local results.

Policy	Half-Brier loss	Accuracy (%)	Training CPU (s)	Models	Predict CPU (ms/1k)
One holdout: fold	0.118153	82.426	32.90	8.35	80.13
One holdout: refit	0.115922	82.832	39.40	8.35	82.98
Select once: fold average	0.116149	82.766	42.47	25.05	240.03
CV: fold average	0.115862	82.841	98.70	29.62	293.54
CV: refit	0.115469	82.860	106.64	9.88	101.63
3 holdouts: fold average	0.115931	82.813	98.70	25.05	240.38
3 holdouts: refit average	0.115513	82.875	112.66	19.28	184.54
3 holdouts: pruned refit	0.115559	82.851	106.76	9.88	101.34

second is below 0.05 while its more conservative simultaneous interval includes zero; we report both constructions. Refitting is not uniformly beneficial: the CV full refit beats the CV fold average by 0.039 [-0.008, 0.087] points, with 12 positive and six negative task means. In an earlier related study, two unregularized boosting fits saturated 2,046 of 2,048 test rows after full refitting and lost to a constant prior, while their L2-regularized siblings did not; CV gave the unstable pair no weight in that draw. The pruned three-holdout average, matched to CV's support size, trails CV by 0.009 [-0.006, 0.024] points. Validation-selected single-configuration baselines reach losses of 0.1210 (holdout, fold), 0.1185 (holdout, refit), 0.1176 (CV, fold average) and 0.1178 (CV, refit), so the ensemble comparison is not an artefact of omitting the usual single-model alternative.

Table C.2. Every local task, in utility points. Matched and Refit are CV minus one holdout before and after common refitting. CV refit minus native compares two CV output constructions. The last two columns compare the CV refit with the ordinary three-holdout refit average and its pruned version. Draws are averaged within task; Shuttle has two draws.

Task	Draws	Matched	Refit	CV refit − native	vs 3H refit	vs pruned
pendigits	8	+0.0176	+0.0168	+0.0947	+0.0021	−0.0018
electricity	8	+0.0639	+0.1893	−0.1592	+0.0038	+0.0235
jm1	8	+0.0423	+0.0542	−0.0091	+0.0175	+0.0260
KDD-appetency	8	+0.0116	+0.0221	−0.0053	+0.0095	+0.0082
Amazon-access	8	+0.0070	+0.0199	+0.0076	−0.0000	+0.0033
PhishingWebsites	8	+0.0303	+0.0288	+0.0878	−0.0259	−0.0236
Diabetes130US	8	+0.0188	+0.0192	+0.0061	−0.0139	−0.0130
numerai28.6	8	+0.0480	+0.0307	−0.0081	+0.0089	−0.0017
connect-4	8	+0.0210	+0.0415	+0.1310	+0.0105	+0.0184
shuttle	2	+0.0075	+0.0364	+0.0365	+0.0144	+0.0088
jungle-chess	8	+0.0179	+0.0233	+0.0648	−0.0068	+0.0003
APSFailure	8	+0.0111	+0.0241	−0.0021	+0.0042	+0.0068
MiniBooNE	8	+0.0327	+0.0685	+0.1317	+0.0336	+0.0995
kick	8	+0.0136	+0.0188	+0.0063	−0.0065	−0.0052
volkert	8	+0.0377	+0.0530	+0.3118	+0.0109	+0.0148
jannis	8	+0.0448	+0.0353	+0.0207	−0.0193	−0.0149
Click-small	8	−0.0155	+0.0882	−0.0317	+0.0216	+0.0197
okcupid-stem	8	+0.0419	+0.0456	+0.0234	+0.0150	−0.0074

Table C.3. The select-once shortcut against four comparators, utility points, one family of four simultaneous intervals. Added after the first local results; nothing was tuned on this outcome.

Shortcut minus	Gain	Simultaneous 95%	Tasks +/−
CV fold average	−0.0287	[−0.0436, −0.0139]	1/17
CV refit	−0.0680	[−0.1279, −0.0081]	1/17
One-holdout refit	−0.0227	[−0.0933, +0.0480]	6/12
Three-holdout fold average	−0.0218	[−0.0404, −0.0033]	2/16

D DEVELOPMENT SPLIT AND CHRONOLOGY

The original development analysis used outer split 0_0 for all 51 tasks. There, averaging and information changed the neural-versus-tree win share by 6.37 [-0.27, 13.01] and 1.96 [-1.60, 5.52] percentage points, and validation-only family selection changed the chosen family on 4 of 51 tasks. The development split is not part of any confirmatory family. An engineering plan was frozen at 15:03 UTC and the final launch plan at 15:12 UTC on 24 September 2026, both before any new-split payload was opened; both are released. The pre-run precision calculation used only development-split task variances. Every registered contrast is reported regardless of sign, and no task, family, metric or stopping rule was chosen from new-split outcomes.