BEYOND ORACLE HEADROOM: EVALUATING CONTEXT SELECTION IN TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Better contexts in hindsight need not yield better decisions at deployment. We study this gap in tabular foundation models using TabPFN and TabICL. Development evaluations show mean Raw Oracle Headroom of 0.318 in negative log-likelihood (NLL), yet expanding a nested candidate bank increases headroom without corresponding gains for the tested selectors. An exact decomposition separates fixed candidate-rule advantage from observed per-query switching opportunity and shows that similar headroom can arise from different candidate-loss structures across datasets. On ten prospectively selected holdout datasets, simple geometric pre-outcome policies beat uniform random choice by 5.69–8.89 percentage points among bank-positive queries. However, post-hoc frequency matching and a direct P2 fixed-rule comparison leave added rule-identity assignment value unestablished. P2 also has exact score ties on 99.80% of queries: tied contexts often share score-determining bottlenecks yet yield different predictions, and its lexicographic tie-break outperforms uniform choice among minima on both models. Once bank predictions are available, predictive centrality (PC) improves over CRUMB in the original formal test, while an advantage over same-bank averaging is not established. On the fresh holdout, gains over averaging are supported only for TabICL, and neither backbone establishes an advantage over CRUMB. Mean gains can also coexist with losses on many individual queries. Together, these results separate hindsight opportunity, policy-level success, and deployment value, showing that context-selection claims depend critically on the comparator and decision stage.

1 INTRODUCTION

Tabular foundation models (TFMs) adapt through labeled support examples, so their predictions depend on the context they receive (Hollmann et al., 2023; 2025; Qu et al., 2025). Contexts can be chosen through retrieval and localization (Thomas et al., 2024; Koshil et al., 2024), distribution matching (Heredge et al., 2026), or optimization and repeated sampling (Feuer et al., 2024; Chen et al., 2026; Abdullah, 2026). With finite labeled contexts, construction and selection determine the task evidence used at deployment. Pre-inference selection can avoid bank inference. Once predictions exist, aggregation can reuse them as a post-bank alternative.

A candidate bank is the set of contexts compared for a query. A hindsight Oracle selects the candidate with the lowest realized loss. A large Oracle advantage can motivate elaborate selectors, but reveals neither the gap's source nor whether pre-outcome information guides choice, rule-identity assignment adds value, or selection beats same-information alternatives. Selecting a minimum among many dependent candidates also complicates statistical interpretation (Cawley & Talbot, 2010; Tibshirani & Tibshirani, 2009; Andrews et al., 2024). Algorithm selection faces a similar distinction: a virtual best solver may outperform a fixed solver even when a selector cannot recover that advantage (Kerschke et al., 2019).

Adding candidates cannot lower hindsight opportunity; bank inference need not improve the ensuing decision.

We ask: when does hindsight context opportunity translate into deployment value in tabular foundation models?

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 decision changes once candidate predictions are available. Single-call policies choose before candidate inference; post-bank rules can select one prediction or combine them. Neither uses query outcomes. CRUMB uses an unlabeled query batch, whereas our single-call policies use the current query. For post-bank selection, probability averaging uses the same predictions without choosing a preferred context. We use predictive centrality (PC) to probe what selection adds to this alternative. Mean utility need not imply query-wise reliability.

TabPFN/TabICL development evaluations show mean Raw Oracle Headroom of 0.318082; nested-bank expansion adds headroom without corresponding P2/P3 gains. Exact $H = G + S$ separates fixed candidate-rule advantage G and observed per-query switching opportunity S, whose relative contributions differ substantially across datasets. Headroom magnitude alone does not reveal candidate-loss organization; the implemented N2 residual-resampling null preserves mean headroom in conditional expectation, leaving stronger residual qualification unresolved. Yet on a prospectively selected 10-dataset fresh holdout, P2 and P3 support pre-outcome identification relative to uniform random candidate choice on both backbones. Supplementary analysis supports positive mean utility over development-selected fixed references, with dataset effects more consistent for TabICL than TabPFN. Post-hoc controls leave added identity-assignment value unestablished. P2's shared score bottlenecks leave contexts with different predictions tied; its lexicographic rule outperforms uniform choice among minima on both models.

The comparison becomes more demanding once bank predictions are available. PC improves on the specified one-context comparators on 18 disjoint OpenML-CC18 replication datasets (Bischl et al., 2021) and a 12-dataset formal test (Erickson et al., 2025), but the formal test does not establish superiority over same-bank averaging. On the fresh holdout, averaging comparisons support a gain for TabICL, while neither backbone establishes a gain over CRUMB. Average gains can also coexist with losses on many individual queries.

Contributions. Across these TFM evaluations, we distinguish sources of context-selection gain and the evidence needed to attribute them. Descriptive $H = G + S$ shows dataset-varying fixed-rule and observed switching opportunity; a fixed rule can still construct query-dependent contexts. On fresh data P2/P3 beat random, but frequency matching and P2's fixed-rule control leave added rule-identity assignment value unestablished; P2's ties show tie-resolution sensitivity. For post-bank PC, one-context gains need not extend to same-bank averaging, and mean gains can coexist with query-level losses.

2 From Hindsight Opportunity to Deployment Value

2.1 Evidence Requirements

Figure 1 separates four questions. **Candidate variation** asks whether contexts offer different outcomes; comparing reference loss with the lowest candidate loss measures hindsight opportunity, not an actionable choice. **Structural interpretation** asks what produces those differences. Claims about residual structure require a specified null and a validated estimator; a large Oracle gap alone is not a structural test. **Pre-outcome identification** asks whether information available when choosing improves selection relative to a stated comparator. Policies must be evaluated without using query outcomes to choose; hindsight-best identities cannot supply this evidence. **Conditional deployment value** asks whether selection improves on practical alternatives given their information and relevant resource constraints. This requires matched outcome comparisons, costs, and query-level harm, not average gain alone. Headroom growth, pre-outcome identification, and deployment gains diverge in our experiments, rather than providing interchangeable evidence.

2.2 Information Timing and Context Actions

P1–P3 choose a context before the model produces any candidate predictions. PC instead selects one prediction from an already computed bank, while probability averaging combines the same predictions. None of these methods uses the query label to make its decision. Section 3 specifies the information and reference available to each method; Appendix B defines the quantities we estimate.

2.3 Comparisons, Resources, and Reliability

For query loss L, paired gain against comparator r is $g_{m:r} = L_r - L_m$, with positive values favoring method m. The corresponding harm is $h_{m:r} = -g_{m:r}$. A method can therefore reduce mean loss while increasing loss on individual queries. We report average gain and query-level harm separately, stating the comparison method, how contexts are built, and their support and compute requirements. Averaging predictions differs from choosing one context; full support also changes the number of examples in that context.

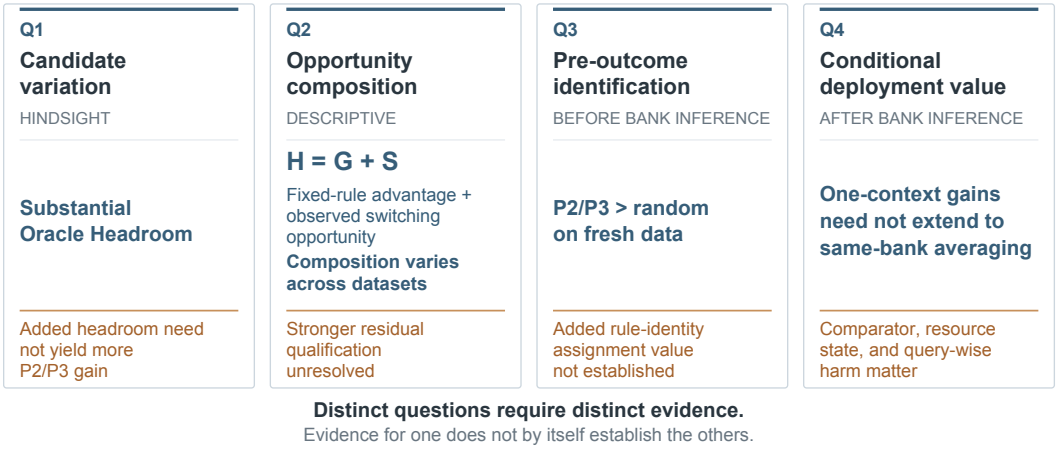


Figure 1: Context-selection evidence from hindsight opportunity to conditional deployment value. Each card pairs an evaluated finding with its interpretation boundary; the questions are distinct, so evidence for one does not by itself establish the others. P1–P3 act before bank inference, whereas PC and averaging use an already computed bank.

3 Experimental Design

3.1 Models, Datasets, and Candidate Contexts

We evaluate fixed TabPFN v2 and TabICL v1 without fine-tuning (Hollmann et al., 2025; Qu et al., 2025). Each unit has separate source-fit, labeled-support, and query partitions. All candidates within a unit share the split, class mapping, and query order. B counts candidates and K support examples per context. The original formal test uses $B = 16$ and $K = 64$. Prespecified B4/B8/B32 checks use nested prefixes of the same ordering; B64 is supplementary. Full support uses all 128 support rows as one context (Appendix A).

3.2 Information Regimes and Context Actions

Single-query pre-inference policies (R0) use query, support, source-side data, and permitted geometry/shift information. P1 selects by query proximity. P2 selects the candidate minimizing the worst query-to-nearest-support distance across source classes, favoring geometric class coverage. P3 uses P2's choice unless its trigger fires, when it uses the reference selected on development data. Each makes one final TFM call. Post-bank rules use all candidate predictions. Neither regime uses query labels, losses, Oracle identities/ranks, or outcome-derived statistics.

For bank predictions $p_1, \ldots, p_B$, PC selects

$$s(c_i) = \frac{1}{B-1} \sum_{j \neq i} \mathrm{JS}(p_i, p_j), \qquad \pi^{\mathrm{PC}} = \arg\min_{c_i} s(c_i),$$

with ties broken by the lowest slot index. Probability averaging combines the same distributions. Both rules require the bank predictions to be computed first.

3.3 EVIDENCE STRATA AND EVALUATION CHRONOLOGY

Development, qualification, exploratory policy evaluation, and prespecified reserve evaluation remain separate. The exploratory pilot motivated PC, specified before evaluation on 18 disjoint OpenML-CC18 replication datasets (Bischl et al., 2021) and 12 formal-test datasets held out from rule development and selection (Erickson et al., 2025). Both use three seeds and 256 queries per unit. MCAR15 masks 15% of query predictor cells. For the original formal test, we specified methods and comparisons before examining outcomes and finalized predictions and selections before accessing query labels.

Designed after formal outcomes, the original supplementary deployment studies reuse the same 12 datasets to characterize scope rather than provide independent confirmation.

We also evaluate selection on a separate 10-dataset fresh holdout. Dataset identities, policies, fixed references, candidate construction, and seeds were fixed before outcome evaluation. The utility aggregation and bootstrap analysis were specified after inference began and are reported as supplementary analysis, separately from the conversion comparison planned in advance (Appendices A.1 and D.11).

3.4 COMPARATORS AND PRACTICAL ALTERNATIVES

R0 gains use the development-selected fixed reference: STANDARDIZED_KNN for TabPFN and RANDOM_SLOT_6 for TabICL, distinct from canonical slot 0. Selection details and the K64-only sensitivity are in Appendix D.8; fresh-holdout reference sensitivity is in Appendix D.11. We compute the expected NLL of uniform-random selection by averaging candidate NLLs; probability averaging combines probabilities before computing NLL. Minimum-entropy selection uses candidate predictions and is a post-bank rule.

SINGLETON_MMD is the strongest one-context comparator in the replication evaluation. The formal comparator, CRUMB, was selected on development data and uses an unlabeled query batch (Heredge et al., 2026), unlike single-query R0. We use the project-local CRUMB implementation described in Appendix F.1, unchanged throughout formal evaluation; we had no verified official implementation. Averaging uses the same bank predictions as PC. Full support uses the same labeled pool with a different context budget. We report original runtimes separately because supplementary runs lack controlled comparisons on matched hardware.

3.5 OUTCOMES, ESTIMANDS, AND UNCERTAINTY

Mean true-class NLL uses natural logarithms and probability clipping at 10^{-12}. Gain is comparator minus method NLL. Within-setting method contrasts and changes in comparative gain differ from direct expansion gain (B32 minus B64 NLL for the same method), which tests whether expansion improves that method.

Raw Oracle Headroom uses a stratum-specific reference and aggregation; the original policy/reserve estimand uses the reference-slot mean (Appendix B). Policy gain instead uses the development-selected fixed reference (BEST_FIXED). A query is bank-positive if any candidate beats this fixed reference; conditional conversion measures how often the policy selects one. The ex-post union counts successes by any policy using outcomes, so it is not a deployable selector.

For the original formal test, we average paired query gains over three seeds within each dataset and backbone, then weight datasets equally. The original 95% intervals use 10,000 paired dataset-cluster bootstrap replicates. Supplementary intervals also average applicable subset repeats and use 10,000 replicates, without replacing original intervals. Backbone-specific results are primary; original secondary pooling retains 12 dataset clusters. Supplementary intervals are pointwise, without multiplicity correction. An interval containing zero establishes neither a direction of effect nor equivalence. Query-harm estimators retain their original definitions (Appendices B and I).

For the fresh holdout, two seeds are aggregated within dataset and ten datasets receive equal weight. Pointwise 95% intervals use 10,000 paired dataset-bootstrap replicates. Conversion conditions on bank-positive queries; policy utility averages fixed-reference NLL minus policy NLL over all queries (Appendix B.5).

3.6 Supplementary characterization and structural interpretation

The original supplementary baseline and B16 construction analyses reuse saved predictions. Identity resampling changes candidate identities while keeping bank composition fixed; high-random also changes that composition. NO_ADDED_MCAR removes only the added mask, retaining native missingness and regenerating query-dependent contexts. B64 adds stratified-random candidates, changing both size and construction. For each paired comparison, predictions for the reference and modified settings come from the same prediction environment. Appendix G details subsets and conditions.

N1 and N2 are diagnostic null constructions. N1 samples labels from a reference predictive distribution while retaining candidate predictions; N2 resamples whole candidate-loss residual vectors. Their empirical contrasts are descriptive; implemented N2 preserves mean headroom in conditional expectation, making its mean-headroom contrast non-identifying. Historical qualification and implementation limits are detailed in Appendix C.5.

4 Candidate Variation and Opportunity Composition

4.1 Substantial Hindsight Variation Within Candidate Banks

The hindsight opportunity is substantial: mean development Raw Oracle Headroom is 0.318082, with a descriptive bootstrap interval of [0.200231, 0.433146], resampling five datasets with the original weighting over 90 unit–backbone records. Mean absolute loss correlation is approximately 0.620 and effective diversity approximately 2.30; Bank-16 does not provide 16 independent opportunities. In a separate four-dataset supplementary analysis, expanding the nested bank from Bank-16 to Bank-32 adds about 0.019–0.031 NLL of headroom, yet P2/P3 conditional conversion falls and paired gain changes little (Appendix C.6). Nesting guarantees non-decreasing headroom, not corresponding gains for these selectors. More opportunity in hindsight need not mean more recoverable value. What, then, does the Oracle gap reveal about candidate-loss structure?

4.2 What Raw Headroom Does and Does Not Reveal

Raw Headroom combines fixed-rule advantage and per-query switching. We decompose it exactly as $H = G + S$: G is the advantage of the best fixed candidate rule over the reference, and S is the additional observed per-query switching opportunity. A fixed rule can generate different physical contexts across queries. Both components describe realized losses, not deployable gains (Appendix C.2).

Equal-dataset means (H, G, S) are $(0.279823, 0.121918, 0.157905)$ for TabPFN and $(0.346569, 0.165215, 0.181354)$ for TabICL. Their composition varies sharply: Mushroom headroom is almost entirely fixed-rule advantage, whereas switching contributes substantially on Adult. Headroom magnitude alone therefore does not determine the opportunity from per-query switching. Limited historical repeats suggest that some query-identity-related advantages persist, but coverage is sparse and pair-level effects are mixed (Appendix C.3).

The decomposition describes opportunity composition. N1/N2 remain construction-specific diagnostics: implemented N2 preserves mean headroom in conditional expectation, so its contrast cannot identify residual structure (Appendix C.5). Whether pre-outcome information guides useful choices is a separate question.

5 Identifying Better Actions Under Information Constraints

5.1 Single-Call Policy-Level Success on the Fresh Holdout

Single-call policies test gains over explicit comparators using pre-outcome information; policy-level success alone does not establish added query-to-rule assignment value.

On 10 prospectively selected fresh-holdout datasets, P2/P3 beat uniform random Bank-16 selection in conditional conversion on both backbones (Figure 2); all four pointwise 95% dataset-bootstrap intervals lie above zero. P1 remains heterogeneous (Appendix D.11).

To understand these gains, we test whether query-to-identity assignment adds value beyond marginal candidate-rule preference. Post-hoc frequency matching preserves identity counts while breaking query–identity correspondence. Three NLL and all conversion intervals cross zero; TabICL P3 NLL is negative (-0.003912, 95% CI $[-0.008325, -0.000473]$). P2 selects CLASS_BALANCED_KNN on 99.22% of queries, uniquely minimizing its score on only 9/5,120. Against fixing this rule, P2's NLL gains are $+0.000112$ $[-0.000519, +0.000950]$ for TabPFN and -0.000317 $[-0.000867, +0.000141]$ for TabICL (pointwise 95% intervals); conditional conversion likewise establishes no positive increment. The fixed rule still constructs query-dependent physical contexts, so this control tests added rule-identity routing, not query conditioning itself.

To explain P2's concentrated identities, we examine its score: P2 minimizes the maximum classwise nearest-support distance, often leaving context differences unresolved. Exact minima remain tied on 5,110/5,120 queries (99.80%), with median minimum-set size eight. Among these tied queries, 93.72% share a bottleneck support row across all minima, despite a pair-weighted median context Jaccard of 0.3474. Yet every tied set contains different predictions and losses on both backbones.

To test whether tie resolution affects outcomes, we hold candidates and scores fixed: the original lexicographic rule beats uniform choice among minima by $+0.036484$ $[+0.013604, +0.063817]$ NLL for TabPFN and $+0.031475$ $[+0.009001, +0.058551]$ for TabICL (post-hoc pointwise 95% dataset-bootstrap intervals); conversion gains are $+12.15/+10.72$ pp. The random comparator averages candidate losses, not predictions.

On the same fresh holdout, post-hoc SmoothMax reduces P2 ties from 99.80% to 88.52% while preserving 99.26% of decisions and similar beyond-random conversion; added rule-identity assignment value remains unestablished (Appendix D.12).

Earlier exploratory/reserve evaluations showed abundant opportunity and heterogeneous utility: P2/P3 beat post-hoc uniform random conversion, but exploratory TabPFN gains were dataset- and extreme-query-sensitive. Because fallback returns the fixed reference, P3 cannot increase the fraction of queries with positive harm relative to P2 by construction; rate-matched random fallback had lower harm rates and mixed tail comparisons, leaving its trigger unvalidated as a risk detector (Appendix D).

Separately, we examine reference sensitivity. Supplementary P2/P3 mean NLL gains over development-selected references are positive on both backbones, each winning 6/10 TabPFN and 9/10 TabICL datasets. Under the K64-only RANDOM_SLOT_6 reference, TabPFN utility and conversion-minus-random intervals remain positive with unchanged selections and P3 fallback (Appendix D.11).

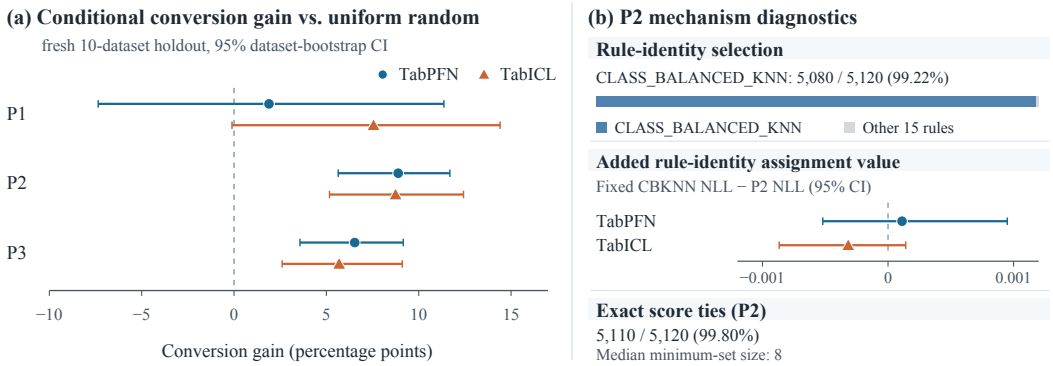


Figure 2: Fresh-holdout performance and P2 diagnostics. (a) Conditional conversion gain over uniform random Bank-16 selection. (b) P2 selects CLASS_BALANCED_KNN on 99.22% of queries; its added NLL gain over that fixed rule is not established, although the rule still constructs query-dependent contexts. Exact score ties occur on 99.80% of queries. All plotted intervals are pointwise 95% dataset-bootstrap intervals. Tie-break sensitivity is reported in the text.

5.2 Selection Once Bank Predictions Are Available

P1–P3 choose before bank inference and can evaluate only their selected context. PC instead acts after bank inference; probability averaging reuses the same predictions, testing select-one value. PC selects

by centrality without training a router and was specified before disjoint replication (Appendix E). An exploratory conservative learned router transferred poorly: leave-one-dataset-out gain was −0.016067 NLL against the pilot's strongest comparator, winning 37.5% of dataset–backbone cells.

6 Deployment Value Relative to Practical Alternatives

6.1 One-Context and Same-Bank Alternatives

CRUMB selects one context from an unlabeled batch without full-bank inference; PC and averaging then use the same bank predictions. In the original 12-dataset MCAR15/Bank-16/K64 formal test, PC gains +0.009311/+0.012325 over CRUMB for TabPFN/TabICL. Against averaging, gains are +0.000675/+0.000054, with both intervals crossing zero (Figure 3). Secondary pooled gains are +0.010818 over CRUMB (original 95% CI [+0.005774, +0.016186]) and +0.000365 over averaging ([−0.000349, +0.001119]). In the original formal test, the supported advantage is over CRUMB, not averaging. At prespecified B4 and B8, TabICL PC underperforms averaging, with original formal-test intervals entirely below zero (Appendix E.4).

The 18-dataset disjoint replication similarly gives PC +0.008832 NLL over SINGLETON_MMD, with positive point estimates on both backbones. Its leave-one-dataset-out range, [+0.003841, +0.011459], is a sensitivity range, not a confidence interval. Gain over averaging is −0.000238; this point estimate alone does not establish inferiority.

On the fresh holdout, neither backbone's interval reproduces PC's original CRUMB advantage; only TabICL's supports PC over same-bank averaging. This comparison uses identical saved candidate predictions and does not depend on the project-local CRUMB implementation (Figure 3; Appendix E.5).

Supplementary comparisons on original B16 predictions show positive pointwise intervals against canonical slot 0, uniform-random expected selection, and minimum entropy for both backbones. Gains over uniform-random expected selection are +0.023109 [+0.016557, +0.030288] for TabPFN and +0.028053 [+0.019465, +0.037971] for TabICL. Averaging remains the critical alternative once bank predictions are available. The complete original and supplementary comparisons are in Appendices E and F.

6.2 Sensitivity to Input Conditions and Candidate-Bank Construction

PC's relative gains vary with input condition and candidate-bank construction. Removing the added MCAR mask retains native missingness and yields a positive averaging comparison for TabPFN but not TabICL; these separate intervals do not test a difference between backbones. For TabICL, the high-random-minus-original change in PC's comparative gain is -0.000844 [-0.001656, -0.000167] against averaging and -0.002510 [-0.004708, -0.000601] against CRUMB. TabPFN's paired-change intervals and both backbones' identity-resampling change intervals cross zero, leaving the directions unresolved. Direct B32-to-B64 gains remain unresolved for PC and averaging; added stratified-random candidates also change composition. TabPFN's B64-versus-CRUMB interval crosses zero. These supplementary comparisons characterize sensitivity within the evaluated settings and constructions (Appendix G).

6.3 Using the Same Support Pool Under a Different Context Budget

Full support uses the same 128-row pool in one K128 context, versus K64 for PC. Under NO_ADDED_MCAR with TabPFN, full support's pointwise gain over PC is $+0.009900$ [$+0.000062$, $+0.023321$]; the other three backbone-by-condition intervals cross zero. This different-budget comparison establishes neither equal-K superiority nor a runtime advantage (Appendix H).

These comparisons concern different deployment decisions. Before candidate inference, the question is whether a single context can be identified from pre-outcome information; after bank predictions exist, selection competes with aggregation of those same predictions; full support instead uses the same support pool under a larger single-context budget. Evidence at one decision stage therefore does not imply superiority at another.

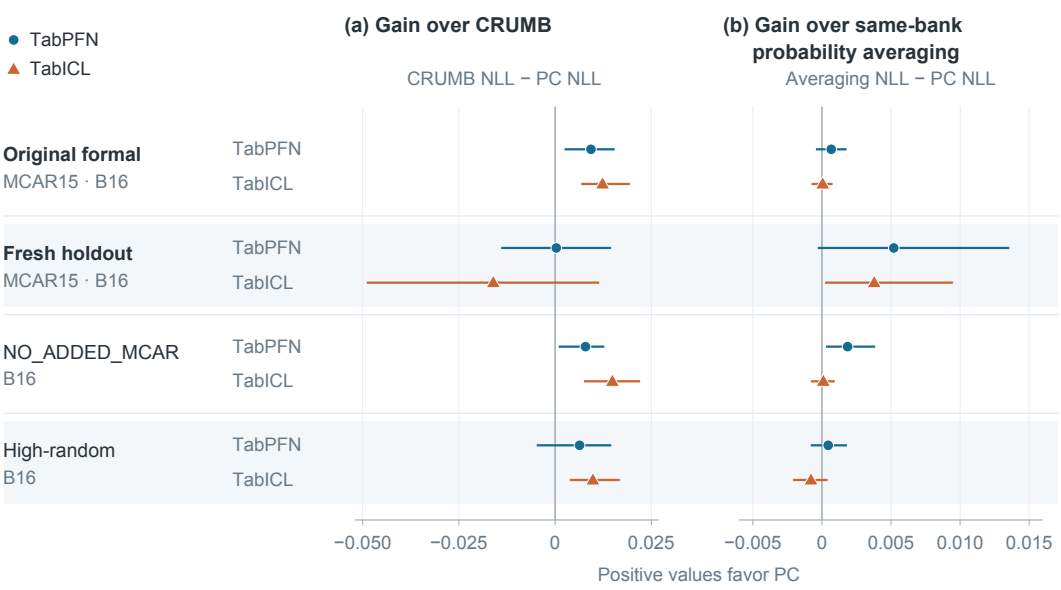


Figure 3: PC's gains depend on the comparison method and setting. The columns compare PC with CRUMB and averaging the same bank predictions; positive gain favors PC. Original formal rows use intervals from the prespecified analysis. Fresh-holdout rows use pointwise 95% paired dataset-bootstrap intervals over ten previously unused datasets. The remaining rows are later supplementary analyses over the original 12 datasets, also with pointwise 95% intervals. Removing the added MCAR mask retains native missing values; the high-random setting changes candidate-bank composition.

6.4 Query-Wise Reliability and Measured Computational Cost

PC's average gains coexist with individual-query losses: original pooled B16 harm rates are 54.71% versus CRUMB and 28.61% versus averaging, with backbone-specific fresh-holdout rates of 58.40–59.30% and 31.21–33.77%, respectively. Loss increases also occur under NO_ADDED_MCAR and B64; Appendix I gives thresholded and tail summaries.

Original B16 used 99.6× physical calls and 103.2× wall-clock time versus one-context CRUMB. Batching and execution choices make these ratios implementation-specific, not intrinsic complexity (Appendix J). PC and averaging reuse bank predictions, testing select-one value after bank inference, though remaining costs may differ. These ratios do not establish B64, full-support, or supplementary runtimes.

7 Related Work

7.1 Context Retrieval, Localization, and Optimization

LUCoS contrasts a supervised validation-set oracle with test-blind, label-free selection before annotation (Ipas et al., 2026); sampling work studies size, stability, coverage, and cost (Abdullah, 2026). Our R0 setting uses labeled support and the current query with a per-query realized-loss bank Oracle. Retrieval, optimization, and acquisition offer other contexts (Thomas et al., 2024; Koshil et al., 2024; Feuer et al., 2024; Treerath & Pittorino, 2026); BAPS compresses prototypes (Jadid et al., 2026), Localized TabICLv2 retrieves in learned space (Guta, 2026), and VIP-COP tests averaging of random contexts (Chen et al., 2026).

7.2 Distribution Matching and Post-Bank Decisions

CRUMB matches context distributions to an unlabeled query batch (Heredge et al., 2026); other work studies test-time shift and adaptation (Lee et al., 2026b; Helli et al., 2024; Lee et al., 2026a). Our development-selected formal comparator is CRUMB; the non-exhaustive panel includes only

an inference-time MIXTUREPFN approximation (Xu et al., 2025). OligoICP selects and averages uncertainty-ranked context-derived predictors for biomolecule regression against full-pool averaging, a full-data model, and a hindsight-best-model reference (Li et al., 2025). PC instead chooses one prediction per query and compares with averaging the same fixed bank.

7.3 Oracle Evaluation, Algorithm Selection, and Query-Level Risk

Multi-LLM routing distinguishes oracle opportunity from pre-answer realizability (Shihab et al., 2026) and tests targeting with share-matched controls (Kumar & Saminathan, 2026). Our controls target added rule-identity assignment despite rule-internal query dependence. Algorithm selection and winner inference caution against reading minima as deployable gains (Kerschke et al., 2019; Cawley & Talbot, 2010; Tibshirani & Tibshirani, 2009; Andrews et al., 2024; Romano & Wolf, 2005). Unlike selective prediction (Geifman & El-Yaniv, 2017; 2019), fallback still predicts; we report paired NLL, query harm, rate-matched fallback, information, and compute.

8 Limitations and Conclusion

8.1 Limitations and Scope of Evidence

The original supplementary studies reuse the 12 formal-test datasets without independent confirmation; their intervals are pointwise and unadjusted for multiple comparisons. Repeating seeds or evaluating another backbone adds no independent datasets, and pooling can hide backbone differences. Datasets held out from rule development may still have been used in backbone pretraining.

The decomposition describes realized development losses: S is not noise-corrected, deployable, or necessarily reproducible. Sparse, dependent repeats in three settings do not establish general query-specific structure. Historical N1/N2 remain construction-specific diagnostics rather than qualified evidence about residual structure.

Fresh-holdout datasets, policies, references, candidate construction, and seeds were fixed before outcome evaluation. Utility aggregation and bootstrap were specified after inference began and remain supplementary. Ten datasets and two backbones do not establish reliability for other banks or conditions. B64 changes bank size and composition together; NO_ADDED_MCAR retains native missingness. Full support changes context size, and the post-bank comparison panel is not exhaustive.

Runtime ratios describe the original implementation. Supplementary runs were not timed on matched hardware, and sharing predictions need not give equal total costs. Harm summaries measure loss increases, not application-specific safety; h>0.05 is an operational threshold rather than a boundary established for a particular domain.

8.2 Conclusion and Implications

In these TFM evaluations, hindsight opportunity, pre-outcome policy success, and deployment value separate empirically. Larger banks add headroom without corresponding P2/P3 gains, and headroom's fixed-rule and observed switching components vary across datasets. Pre-outcome policies beat uniform random choice, yet frequency-matched and fixed-rule controls leave added P2 assignment value unestablished. P2's shared score bottlenecks leave different predictions tied, and its lexicographic rule outperforms random choice among minima. Positive TabPFN utility persists under the specified K64-only reference sensitivity. Post-bank one-context gains need not extend to averaging; mean gains coexist with individual-query losses.

Oracle Headroom is therefore an opportunity diagnostic, not evidence of recoverable value. For the evaluated geometric policies, score discriminability, tie-breaking sensitivity, and identity-assignment value require separate interpretation. Context-selection evidence depends on the decision stage: information, alternatives, incurred resources, and query-wise outcomes.

AI USE STATEMENT

The authors led the research questions, conceptual framing, experimental design, figure design, interpretation of results, manuscript organization, and final scientific judgments. Generative AI tools, including ChatGPT and Codex, assisted with methodological brainstorming, code development and debugging, literature search and synthesis, interpretation and presentation of results, drafting portions of the manuscript, and manuscript revision and language editing. The authors independently evaluated, edited, and verified all AI-assisted suggestions and outputs before including them. The authors take full responsibility for the experiments, analyses, figures, citations, scientific claims, and conclusions.

REPRODUCIBILITY STATEMENT

Sections 2–3 and Appendices A–J describe the data, models, context construction, selection rules, statistical definitions, and implementation details. Appendices A and E list the disjoint replication and formal-test datasets, seeds, and order of evaluation. For the original replication and formal evaluations, methods, comparators, metrics, and analysis code were specified before test evaluation; labels were accessed only after predictions and selections were finalized. Historical post-hoc analyses use saved outputs where stated. The fresh-holdout analyses in Appendices D.11 and E.5 use newly generated predictions, with datasets, policies, fixed references, candidate construction, and inference seeds fixed before outcome evaluation. The later P2/P3 utility aggregation and bootstrap summary are reported as supplementary analysis.

We later compared the intended structural-qualification protocol with its historical implementation. This review identified implementation differences and components not uniquely specified for an end-to-end corrective rerun. No corrected qualification simulation was run. We report the historical N1/N2 analyses as diagnostic, not as qualified evidence for or against residual structure; this later review is not part of the original prespecification.

REFERENCES

Mohammed Abdullah. Understanding context sampling in TabPFN on small tabular datasets. arXiv preprint arXiv:2607.26628, 2026.

Isaiah Andrews, Toru Kitagawa, and Adam McCloskey. Inference on winners. *The Quarterly Journal of Economics*, 139(1):305–358, 2024. doi: 10.1093/qje/qjad043.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael G. Mantovani, Jan N. van Rijn, and Joaquin Vanschoren. OpenML benchmarking suites. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Jock Blackard. Covertype. UCI Machine Learning Repository, 1998.

R. Bock. MAGIC gamma telescope. UCI Machine Learning Repository, 2004.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008. doi: 10.1162/rest.90.3.414.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010.

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

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018. doi: 10.1111/ectj.12097.

Ilkay Cinar and M. Koklu. Rice (Cammeo and Osmancik). UCI Machine Learning Repository, 2019.

Ilkay Cinar, Murat Koklu, and Sakir Tasdemir. Raisin. UCI Machine Learning Repository, 2020.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Wine quality. UCI Machine Learning Repository, 2009.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025.

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*, volume 37, 2024. doi: 10.52202/079017-2654.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019.

Beimnet Bekele Guta. Localized TabICLv2: Scaling tabular in-context learning through k-NN. In *2nd ICML Workshop on Foundation Models for Structured Data*, 2026.

Kai Helli, David Schnurr, Noah Hollmann, Samuel Müller, and Frank Hutter. Drift-resilient TabPFN: In-context learning temporal distribution shifts on tabular data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3134.

Jamie Heredge, Mattia J. Villani, Pranav Deshpande, Akshay Seshadri, and Niraj Kumar. CRUMB: Efficient prior fitted network inference via distributionally matched context batching. arXiv preprint arXiv:2606.11473, 2026.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 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. doi: 10.1038/s41586-024-08328-6.

Mark Hopkins, Erik Reeber, George Forman, and Jaap Suermondt. Spambase. UCI Machine Learning Repository, 1999.

Oroel Ipas, Guillermo Gomez-Trenado, Rocío Romero-Zaliz, and Isaac Triguero. LUCoS: Latent unsupervised context selection for tabular foundation models. arXiv preprint arXiv:2605.27254, 2026.

Mahboobe Jadid, Melika Rezaye Garkani, and Ali Mousavi. Balanced adaptive prototype selection for scalable TabPFN inference on large-scale tabular data. arXiv preprint arXiv:2608.12989, 2026.

Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart disease. UCI Machine Learning Repository, 1989.

Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary Computation*, 27(1):3–45, 2019. doi: 10.1162/evco_a_00242.

M. Koklu and Ilker Ali Özkan. Dry bean. UCI Machine Learning Repository, 2020.

Mykhailo Koshil, Thomas Nagler, Matthias Feurer, and Katharina Eggensperger. Towards localization via data embedding for TabPFN. In *Third Table Representation Learning Workshop at NeurIPS 2024*, 2024.

Kiran N. Kumar and Santhosh K. Saminathan. Task- and session-level model routing: A common-interface hybrid evaluation of four open-source routers across four benchmarks. arXiv preprint arXiv:2608.14641, 2026.

Jinmo Lee, Doyun Choi, Moongi Choi, and Jaemin Yoo. Bounded context management for tabular foundation models on stream learning. In *2nd ICML Workshop on Foundation Models for Structured Data*, 2026a.

Seunghan Lee, Jaehoon Lee, Jun Seo, Sungdong Yoo, Minjae Kim, Taeyoon Lim, Dongwan Kang, Hwanil Choi, Soonyoung Lee, and Wonbin Ahn. Mitigating label shift in tabular in-context learning via test-time posterior adjustment. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2026b.

Jie Li, Andrew McCarthy, Zhizhuo Zhang, and Stephen Young. Uncertainty-guided model selection for tabular foundation models in biomolecule efficacy prediction. In *2nd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences at NeurIPS 2025*, 2025.

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

Volker Lohweg. Banknote authentication. UCI Machine Learning Repository, 2012.

Robert Lyon. HTRU2. UCI Machine Learning Repository, 2015.

James G. MacKinnon and Matthew D. Webb. Wild bootstrap inference for wildly different cluster sizes. *Journal of Applied Econometrics*, 32(2):233–254, 2017. doi: 10.1002/jae.2508.

Rami Mohammad and Lee McCluskey. Phishing websites. UCI Machine Learning Repository, 2012.

Sérgio Moro, Paulo Rita, and Paulo Cortez. Bank marketing. UCI Machine Learning Repository, 2014.

Belinda Phipson and Gordon K. Smyth. Permutation P-values should never be zero: Calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article 39, 2010. doi: 10.2202/1544-6115.1585.

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 *Proceedings of Machine Learning Research*, pp. 50817–50847, 2025.

Valentim Realinho, Mónica Vieira Martins, Jorge Machado, and Luís Baptista. Predict students' dropout and academic success. UCI Machine Learning Repository, 2021.

Joseph P. Romano and Michael Wolf. Exact and approximate stepdown methods for multiple hypothesis testing. *Journal of the American Statistical Association*, 100(469):94–108, 2005. doi: 10.1198/016214504000000539.

C. Sakar and Yomi Kastro. Online shoppers purchasing intention dataset. UCI Machine Learning Repository, 2018.

Jeffrey Schlimmer. Mushroom. UCI Machine Learning Repository, 1981.

Ibne Farabi Shihab, Abu Sa-Adat Mohamed Moon-Im Al Ahsan, and Md Najmus Swaqeeb. Opportunity is not realizability: Selection-valid diagnostics for multi-LLM routing. arXiv preprint arXiv:2608.08265, 2026.

David Slate. Letter recognition. UCI Machine Learning Repository, 1991.

Ashwin Srinivasan. Statlog (landsat satellite). UCI Machine Learning Repository, 1993.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L. Caterini. Retrieval & fine-tuning for in-context tabular models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3442.

Ryan J. Tibshirani and Robert Tibshirani. A bias correction for the minimum error rate in cross-validation. *The Annals of Applied Statistics*, 3(2):822–829, 2009. doi: 10.1214/08-AOAS224.

Wilailuck Treerath and Fabrizio Pittorino. Active in-context learning for tabular foundation models. arXiv preprint arXiv:2603.27385, 2026.

William Wolberg, Olvi Mangasarian, Nick Street, and W. Street. Breast cancer wisconsin (diagnostic). UCI Machine Learning Repository, 1993.

Derek Xu, Olcay Cirit, Reza Asadi, Yizhou Sun, and Wei Wang. Mixture of in-context prompters for tabular PFNs. In *The Thirteenth International Conference on Learning Representations*, 2025.

I-Cheng Yeh. Default of credit card clients. UCI Machine Learning Repository, 2009.

A Experimental Details and Candidate Construction

A.1 Evidence Strata and Datasets

Each evaluation has a separate purpose and dataset allocation. The development analysis contains 45 units from five datasets; null qualification uses 30 units from three other datasets. The exploratory single-call evaluation uses 40 units from four datasets, and the reserve evaluation uses 20 units from four datasets. The earlier bank-size analysis uses another four-dataset set. PC is evaluated on 18 OpenML-CC18 replication datasets and then on 12 TabArena/OpenML formal-test datasets. The original supplementary deployment analyses reuse the 12 formal-test datasets and their three seeds.

Fresh-holdout design. Table 1 lists ten OpenML datasets not used in any earlier evaluation, in the order selected before evaluation (Bischl et al., 2021). We use our prespecified source-fit, support, and query split rather than official OpenML task splits. Each dataset is evaluated with TabPFN v2 and TabICL v1 using seeds 2026091801 and 2026091802, 128 source-fit rows, 128 labeled support rows, and 256 queries per seed. The setting is MCAR15/Bank-16/K64, with the original candidate construction and P1–P3 rules. The development-selected fixed references remain STANDARDIZED_KNN for TabPFN and RANDOM_SLOT_6 for TabICL; these identities were selected before fresh-holdout evaluation. Datasets, policies, references, candidate construction, and inference seeds were fixed before outcome evaluation; references were not reselected, and no dataset was replaced based on results.

Table 1: Fresh-holdout datasets, listed in the order selected before evaluation.

OpenML ID	Dataset
1063	kc2
1049	pc4
1050	pc3
469	analcatdata_dmft
54	vehicle
1480	ilpd
32	pendigits
6332	cylinder-bands
40701	churn
28	optdigits

A.2 Models and Data Partitions

We use source-fit observations to estimate the preprocessing parameters and source-side quantities specified in the design. Labeled support observations provide the examples used to build candidate contexts. Query labels are used only to evaluate outcomes, after predictions and selections have been finalized. The replication and original formal test use 128 source-fit rows, a 128-row labeled support pool, and 256 queries per dataset and seed. Earlier evaluations use the partition sizes listed in their design tables. Replication seeds are 813786334, 2074888709, and 2612954880; formal-test and supplementary seeds are 2452511701, 4162428205, and 1281371069. The complete design table retains the five-seed allocations for the earlier strata.

The evaluated backbones are TabPFN v2 and TabICL v1. The original specifications identify package versions 8.0.8 and 2.1.1 and checkpoints `tabpfn-v2-classifier-finetuned-zk73skhh.ckpt` and `tabicl-classifier-v1-20250208.ckpt`, respectively. Runtime details for the original and supplementary evaluations are reported separately where they differ. The dataset split separates rule development from evaluation. It does not establish that the datasets were absent from backbone pretraining. OpenML 46960 in the formal test is another repository version of UCI 697, used in the older bank-size sensitivity. That earlier sensitivity did not inform predictive-centrality rule development, comparator selection, or formal-test design.

A.3 Candidate Identities and Context Budgets

We use B for the number of candidate contexts and K for the number of support rows in each context. The original B4, B8, B16, and B32 banks use the first 4, 8, 16, and 32 candidates from the same fixed ordering. The B64 characterization preserves slots 0–31 and adds 32 stratified-random candidates. Predictions for its B16 and B32 reference settings come from the same supplementary prediction environment. These prefixes are distinct from the B16 subset constructions used to vary candidate-bank composition. The full-support alternative puts the same 128 labeled support examples into one context, in their original row order, giving B1/K128.

A.4 Operating Conditions

Table 6 specifies candidate construction rules, family membership, preprocessing, seeds, slot identities, and bank-prefix membership. Canonical slot 0 is distinct from the development-selected fixed reference used by P1–P3: STANDARDIZED_KNN for TabPFN and RANDOM_SLOT_6 for TabICL. MCAR15 adds missingness to 15% of query predictor cells. NO_ADDED_MCAR removes only that added mask and retains native missingness; query-dependent contexts are constructed for the corresponding query inputs.

A.5 Complete Dataset, Shift, and Candidate Definitions

The definitions below retain the original experiment names and describe the implemented constructions, including differences from the planned design.

Development, qualification, and exploratory units use seeds 2620110859, 3047047959, 3143979043, 3353531537, and 4053111363. Reserve units use 193864969, 3057041973, 3168119444, 4055826012, and 3454006291. Policy distances are standardized using source-fit data. Candidate construction uses the fitting partition specified for each rule, including support-fitted standardization where stated. Dataset sources follow the existing UCI records (Becker & Kohavi, 1996; Koklu & Özkan, 2020; Schlimmer, 1981; Cinar & Koklu, 2019; Hopkins et al., 1999; Yeh, 2009; Slate, 1991; Bock, 2004; Moro et al., 2014; Blackard, 1998; Sakar & Kastro, 2018; Cortez et al., 2009; Wolberg et al., 1993; Janosi et al., 1989; Mohammad & McCluskey, 2012; Cinar et al., 2020). Replication and formal identities follow OpenML-CC18 and TabArena (Bischl et al., 2021; Erickson et al., 2025).

Table 2: Complete development and qualification design. Every listed cell uses five fixed seeds and both backbones.

Role	Dataset	Shift cells	Source fit	Support pool	Queries
Development	Adult	Class-prior 80/20 reverse; covariate or nuisance $\delta 2$	1,024	768–2,048	256
Development	Dry Bean	Controlled interaction median reflect; density/mixture 80/20 reverse	1,024	768–2,048	256
Development	Mushroom	Class-prior 80/20 reverse; covariate or nuisance $\delta 2$	1,024	768–2,048	256
Development	Rice Cammeo/Osmancik	Controlled interaction median reflect	1,024	768–2,048	256
Development	Spambase	Covariate or nuisance $\delta 2$; local-support central hole	1,024	768–2,048	256
Qualification	Credit Default	Class-prior 80/20 reverse; local-support central hole	1,024	2,048	256/512
Qualification	Letter Recognition	Controlled interaction median reflect; density/mixture 80/20 reverse	1,024	2,048	256/512
Qualification	MAGIC Gamma	Density/mixture 80/20 reverse; local-support central hole	1,024	2,048	256/512

Table 3: Complete exploratory and pre-specified reserve design. Exploratory rows use five seeds per dataset-shift cell; reserve rows use five seeds per dataset.

Role	Dataset	Shift cells	Source fit	Support pool	Queries
Exploratory	Bank Marketing	Covariate or nuisance $\delta 2$; density/mixture 80/20 reverse	1,024	2,048	256
Exploratory	Covertype	Local-support central hole; natural geographic Area 2	1,024	2,048	256
Exploratory	Online Shoppers	Class-prior 80/20 reverse; natural temporal early/late month	1,024	2,048	256
Exploratory	Wine Quality	Controlled interaction median reflect; covariate or nuisance $\delta 2$	1,024	2,048	256
Reserve	Breast Cancer Diagnostic (UCI 17)	Query-side MCAR 15%	128	128	256
Reserve	Heart Disease Four-Cohort (UCI 45)	Query-side MCAR 15%	128	128	256
Reserve	Phishing Websites (UCI 327)	Query-side MCAR 15%	128	128	256
Reserve	Raisin (UCI 850)	Query-side MCAR 15%	128	128	256

Table 4: Datasets used for PC replication and for the formal test held out from rule development and selection. OpenML IDs distinguish datasets with similar names.

Disjoint OpenML-CC18 replication		Formal TabArena/OpenML test	
ID	Dataset	ID	Dataset
11	balance-scale	46921	diabetes
14	mfeat-fourier	46956	seismic-bumps
23	cmc	46938	Is-this-a-good-customer
31	credit-g	46922	Diabetes130US
38	sick	46960	students dropout and academic success
50	tic-tac-toe	46916	coil2000 insurance policies
188	eucalyptus	46969	NATICUSdroid
334	monks-problems-2	46911	Bank Customer Churn
1067	kc1	46940	Marketing Campaign
1068	pc1	46941	maternal health risk
1475	first-order-theorem-proving	46958	splice
1487	ozone-level-8hr	46913	blood-transfusion-service-center
1489	phoneme		
1494	qsar-biodeg		
40975	car		
40983	wilt		
40984	segment		
41027	jungle chess 2pcs raw endgame complete		

Table 5: Shift definitions for the evaluations. All shift parameters are fitted before inference using source-fit information; query labels and query statistics are not used to fit those parameters.

Shift	Fixed construction	Information boundary and purpose
Class-prior reverse	Binary-only source-fit majority/minority quotas; source 80/20 and query 20/80	Quotas are fit from source labels; query labels are not used for parameters and only the fixed quota is applied at generation.
Covariate/nuisance $\delta 2$	Append deterministic synthetic $a2_nuisance_z$ as an additional model-input column after the dataset's existing features; source/support target $N(0,1)$ and query target $N(2,1)$	Generated from unit, fixed seed, row identifier, and partition role without query labels or raw feature values; the column is included in the model input and isolates the specified nuisance shift.
Median reflect interaction	First three eligible numeric features with source-fit medians m_1, m_2, m_3; reflect feature 1 about m_1 when $x_3 \leq m_3$, and feature 2 about m_2 when $x_3 > m_3$	Support is unchanged and labels are unused; all feature choices and boundaries are source-fit and lexicographic.
Density/mixture reverse	Implemented rule: B if the first eligible numeric feature is observed and exceeds its source-fit median, A otherwise; support A/B quotas 80/20 and query quotas 20/80	The implementation uses one median indicator, rather than the specified two-indicator construction; no indicator-disagreement branch is applied.
Local-support central hole	First eligible numeric feature; source-fit rank; query rows in $[0.45, 0.55]$ and support excludes $[0.35, 0.65]$	Deterministic boundary ties; support membership is fixed before query outcome access.
Natural geographic	Covertype Area 2/Neota target and Areas 1,3,4 source	Uses the real wilderness-area field and the deterministic second official area rule; all target classes occur in source.
Natural temporal	Online Shoppers February–August source and September–December target	Uses the real time field; the chronological split is fixed before inference.
Reserve MCAR15	Mask 15% of query predictor cells only	Source/support data, identifiers, targets, and native missingness are preserved; the mask is applied after the source-fit protocol is fixed.

Table 6: Panel A: Base Bank-16 candidate definitions for R0. Each identity is prefixed by `tabicl_v1__R0__` or `tabpfn_v2__R0__`; suffixes and order are shared across backbones. The banks of 4, 8, 16, and 32 candidates take prefixes of this prespecified ordering.

Rank	Identity suffix	Family	Fixed construction or seed	Prefix membership
0	`REFERENCE_SLOT_0`	Canonical reference	Reference seed 101	Bank -4/8/16/32
1	`RANDOM_SLOT_0`	Stratified random	Random seed 401	Bank -4/8/16/32
2	`RAW_KNN`	Raw query geometry	Euclidean raw-feature distance; stable argsort; first K	Bank -4/8/16/32
3	`STANDARDIZED_KNN`	Standardized query geometry	Support-fitted standardization; Euclidean distance; stable argsort; first K	Bank -4/8/16/32
4	`CLASS_BALANCED_KNN`	Label-aware support	Support-fitted query geometry; equal class allocation to nearest rows per class, then support-order fill.	Bank-8/16/32

Continued on next page

Table 6 – continued from previous page

Rank	Identity suffix	Family	Fixed construction or seed	Prefix membership
5	LOCAL_ FARTHEST_ FIRST	Query-local diversity	Nearest $4K$ standardized pool; deterministic farthest-first	Bank-8/16/32
6	GLOBAL_ COVERAGE	Global support coverage	MiniBatchKMeans K clusters; nearest unique support points; seed offset 7001	Bank-8/16/32
7	SINGLETON_MMD	Query -singleton MMD	Standardized singleton-query kernel-MMD greedy rule; nearest $4K$ pool	Bank-8/16/32
8	REFERENCE_ SLOT_1	Canonical reference	Reference seed 211	Bank-16/32
9	REFERENCE_ SLOT_2	Canonical reference	Reference seed 307	Bank-16/32
10	RANDOM_SLOT_1	Stratified random	Random seed 409	Bank-16/32
11	RANDOM_SLOT_2	Stratified random	Random seed 419	Bank-16/32
12	RANDOM_SLOT_3	Stratified random	Random seed 421	Bank-16/32
13	RANDOM_SLOT_4	Stratified random	Random seed 431	Bank-16/32
14	RANDOM_SLOT_5	Stratified random	Random seed 433	Bank-16/32
15	RANDOM_SLOT_6	Stratified random	Random seed 439	Bank-16/32

Table 6: Panel B: Candidates added for the prespecified Bank-32 extension. Slots 16–31 are stratified-random candidates, chosen without evaluation outcomes using the deterministic seeded construction described in Appendix A.

Rank	Identity suffix	Random seed	Prefix membership
16	RANDOM_SLOT_7	443	Bank-32
17	RANDOM_SLOT_8	449	Bank-32
18	RANDOM_SLOT_9	457	Bank-32
19	RANDOM_SLOT_10	461	Bank-32
20	RANDOM_SLOT_11	463	Bank-32
21	RANDOM_SLOT_12	467	Bank-32
22	RANDOM_SLOT_13	479	Bank-32
23	RANDOM_SLOT_14	487	Bank-32
24	RANDOM_SLOT_15	491	Bank-32
25	RANDOM_SLOT_16	499	Bank-32
26	RANDOM_SLOT_17	503	Bank-32
27	RANDOM_SLOT_18	509	Bank-32
28	RANDOM_SLOT_19	521	Bank-32
29	RANDOM_SLOT_20	523	Bank-32
30	RANDOM_SLOT_21	541	Bank-32
31	RANDOM_SLOT_22	547	Bank-32

Distances with missing query values. For reserve MCAR15 and the original formal test, we impute missing values before routing or constructing contexts. Numeric columns use source-fit medians, followed by scaling with the source-fit mean and population standard deviation; constant columns use unit scale. Categorical columns use the source-fit mode, with lexicographic tie-breaking. We assign distinct codes to an entirely missing source column and to an unseen category.

P1 and P2 compute the square root of the mean of squared numeric differences and categorical mismatch indicators over all coordinates. RAW_KNN uses Euclidean distance on the preprocessed

inputs. STANDARDIZED_KNN, CLASS_BALANCED_KNN, LOCAL_FARTHEST_FIRST, SINGLETON_MMD, and GLOBAL_COVERAGE additionally standardize using the support data. None uses pairwise deletion or renormalizes distances by the number of observed coordinates. These implementations reject non-finite feature inputs instead of substituting a distance. The earlier bank-scaling study uses a different imputation step: after categorical encoding, it replaces non-finite entries with source-fit medians, or zero when no finite median exists, and then applies source-fit scaling before computing distances.

Local candidate construction. Distance sorts are stable, so equal distances retain support-row order. LOCAL_FARTHEST_FIRST and SINGLETON_MMD form a query-dependent pool of the nearest $\min(4K, n_{\text{support}})$ rows after support-fitted standardization. Farthest-first starts at the nearest row to the query, then repeatedly maximizes the minimum distance to the selected rows until K distinct rows have been selected; ties take the earliest row in pool order.

SINGLETON_MMD uses the Gaussian kernel $k(x, z) = \exp[-\|x - z\|^2/(2\sigma^2)]$. Its bandwidth is the median of strictly positive pairwise distances among the first $\min(256, n_{\text{pool}})$ pool rows, bounded below by 10^{-6}; if all distances are zero, it uses $\sigma = 1$. Starting with an empty selected set S, at step $t = |S|$ it chooses an unselected row maximizing $2k(x, q) - \frac{2}{t+1}\sum_{z\in S} k(x, z)$. It stops after exactly K selections, with ties resolved by pool order.

CLASS_BALANCED_KNN visits classes in sorted label order and takes up to $\lfloor K/C \rfloor$ nearest rows per class, where C is the number of support classes. Any positions left by remainders or class shortages are filled in ascending support-index order, not by distance, because the set-difference operation sorts the unused indices. P1 breaks candidate ties by nearest distance, then mean distance, then lexicographic candidate identity; P2 breaks coverage-score ties by lexicographic candidate identity, rather than bank slot rank.

The B64 extension uses stratified-random seed offsets 2026091532–2026091563 for slots 32–63. Construction subset seeds are 2026091501–2026091510. These seeds apply to the supplementary constructions, not to the original candidate bank.

A.6 Information Timing and Context Actions

Table 7: Information available when each method makes its choice, and how it uses the context. A context evaluation need not correspond to one physical API call. No method uses the query outcome to make the choice.

Action	Decision information	Context use and role
BEST_FIXED	Development-selected identity	One K64 context; R0 reference
Slot 0	Preset bank identity	One K64 context; simple baseline
P1–P3	Query, support, source geometry	One selected K64 context; identification
SINGLETON_MMD	Support and singleton query	One K64 context; replication comparator
CRUMB	Support and unlabeled query batch	One K64 context per constructed context; formal comparator
PC	All candidate predictions	B contexts; select one prediction
Averaging	Same candidate predictions	B contexts; aggregate probabilities
Full support	Entire labeled support pool	One K128 context; different-budget alternative

B Estimands, Aggregation, and Statistical Details

B.1 Loss, Headroom, and Comparative Gains

A unit u specifies a dataset, a particular shift cell, and a seed. The backbone is indexed separately by b, and q identifies a query within the unit. Candidate contexts and policies are indexed by c and j. A

shift cell denotes the specific shift instance, not only its family. Let $L_{ubq}(m) = -\log(\text{clip}(p_m(y_{uq} \mid x_{uq}), 10^{-12}, 1))$ denote query NLL. For method m and comparator r,

$$g_{ubq}^{m:r} = L_{ubq}(r) - L_{ubq}(m), \qquad h_{ubq}^{m:r} = L_{ubq}(m) - L_{ubq}(r) = -g_{ubq}^{m:r}.$$

Positive gain favors the evaluated method. Policy gain uses the development-selected fixed reference; PC gain uses the comparator named in its evaluation.

We compute Raw Oracle Headroom using the reference and averaging rule specified for each analysis:

$$H_{ubq}^{\text{raw}} = L_{ubq}^{\text{raw,ref}} - \min_{c \in \mathcal{C}_{ubq}} L_{ubq}(c).$$

In the original policy and reserve estimand, $L_{ubq}^{\text{raw,ref}} = \frac{1}{3}\sum_{c \in \mathcal{R}} L_{ubq}(c)$ for reference slots $\mathcal{R} = \{0, 8, 9\}$. This reference-slot mean averages losses, not probabilities, and differs from the development-selected fixed reference used for policy gains. The empirical N1 label-generating distribution instead averages and renormalizes the reference probability vectors.

The prespecified null-specific targets are

$$H_m^{\text{adj}} = \overline{H}^{\text{raw}} - \mathbb{E}_m[\overline{H}^{\text{raw}}], \qquad F_m = \frac{T - \mathbb{E}_m[T]}{D},$$

where T measures squared loss after double-centering to remove additive query and candidate effects, with weights for duplicate groups. In this prespecified definition, candidate effects are fitted on training-query folds. D uses the corresponding weights to measure squared deviations from the unit grand mean on the same bank and query set. These targets are computed within unit and averaged equally over units for each backbone, retaining negative values. The structure fraction F_m is not a headroom-recovery ratio. The implemented qualification simulation instead generated scalar endpoint scores. The biases in Table 12 compare these scores with their assigned scalar truths, rather than evaluating the complete loss-surface estimators.

Write $\mathcal{A}_s$ for the specified aggregation within setting s. Comparative gain and its change are

$$G_{m:r}(s) = \mathcal{A}_s[g^{m:r}], \qquad \Delta G_{m:r} = G_{m:r}(s_1) - G_{m:r}(s_0).$$

A negative change means that the gain over the comparator decreases. Direct expansion gain compares the same method on matched queries, using B32 and B64 predictions from the same prediction environment:

$$E_m = \mathcal{A}[L_{m,B32} - L_{m,B64}].$$

Positive E_m favors the expanded bank. It is distinct from $G_{\text{PC:AVG}}(B64)$. Full-support gain against PC is $\mathcal{A}[L_{\text{PC}} - L_{\text{FULL}}]$, with positive values favoring full support.

B.2 Opportunity, Conversion, and Ex-Post Union

Let L_{ubq}^0 be the loss of the development-selected fixed reference and $L_{ubq}^\star = \min_{c \in \mathcal{C}_{ubq}} L_{ubq}(c)$. Then

$$\delta_{ubq}^{\text{oracle}} = L_{ubq}^0 - L_{ubq}^\star, \quad O_{ubq} = \mathbf{1}[\delta_{ubq}^{\text{oracle}} > 0], \quad R_{ubqj} = \mathbf{1}[L_{ubq}(\pi_j) < L_{ubq}^0].$$

Opportunity prevalence uses all query–backbone observations in the relevant evidence stratum. Conditional conversion and policy-union coverage are

$$\text{Conv}_j = \frac{\sum_{ubq} O_{ubq} R_{ubqj}}{\sum_{ubq} O_{ubq}}, \qquad \text{Union} = \frac{\sum_{ubq} O_{ubq} \max_j R_{ubqj}}{\sum_{ubq} O_{ubq}}.$$

Conversion asks how often a policy selects a context that beats the reference, among queries where such a context exists. The ex-post union counts successes by any evaluated policy after observing outcomes; it does not tell us which policy to choose before prediction. Threshold sensitivity uses $O_{ubq}(\varepsilon) = \mathbf{1}[\delta_{ubq}^{\text{oracle}} > \varepsilon]$, retaining $\varepsilon = 0$ as primary.

B.3 Aggregation and Uncertainty

For the development summary, we average queries within each unit–backbone record, then give the 90 records equal weight. The descriptive interval uses 5,000 two-stage bootstrap repetitions: datasets are sampled first, followed by units, across the five datasets. This weighting is not an equal-dataset mean. Exploratory policy results also give units equal weight. Reserve results instead average queries within units, seeds within cells, cells within datasets, and finally the four datasets equally. Their descriptive intervals use the original 5,000-repetition two-stage bootstrap.

The original formal intervals use the prespecified 10,000 paired dataset-cluster percentile bootstrap repetitions, averaging matched query effects over three seeds within each dataset and backbone before weighting datasets equally. Original supplementary intervals use 10,000 paired dataset-cluster percentile bootstrap repetitions, with three seeds averaged within dataset and backbone. Candidate-subset repetitions are also averaged within dataset before resampling. Datasets receive equal weight. The secondary pooled analysis first averages backbones within dataset. It therefore contains 12 independent dataset clusters, not 24 independent dataset–backbone clusters. Supplementary intervals are pointwise without multiplicity correction. The fresh-holdout definitions are given in Appendix B.5.

A LODO range shows how the mean changes when each dataset is omitted; it is not a confidence interval. W/T/L counts retain their reported units: original formal pooled counts use 24 dataset–backbone cells, while supplementary pooled counts use dataset-level pooled effects. Counts are descriptive. An interval crossing zero establishes neither a direction nor equivalence. When analyses give the same point estimate but different bootstrap intervals, we retain each analysis's own interval.

B.4 Reliability Estimands

For a specified matched-harm pool $\mathcal{H} = \{h_i\}_{i=1}^n$, report $n^{-1}\sum_i \mathbf{1}[h_i > 0]$, $n^{-1}\sum_i \mathbf{1}[h_i > 0.05]$, and P95. With empirical 90th percentile $q_{0.90}(\mathcal{H})$,

$$T_{0.90}(\mathcal{H}) = \frac{\sum_i h_i \mathbf{1}[h_i \geq q_{0.90}(\mathcal{H})]}{\sum_i \mathbf{1}[h_i \geq q_{0.90}(\mathcal{H})]}.$$

This formal/post-bank tail mean includes ties. If $h_{(1)} \leq \cdots \leq h_{(n)}$ and $k = \lceil 0.10n \rceil$, the original single-call tail definition is

$$\mathrm{CVaR90}_{\mathrm{single}} = \frac{1}{k}\sum_{i=n-k+1}^{n} h_{(i)}.$$

The two estimators can differ when values are tied at the threshold. Appendix I specifies which observations are pooled. The threshold 0.05 is operational, not an application-specific harm boundary. Averaging candidate losses gives the expected loss of random selection, not the harm distribution from drawing one random context per query.

B.5 Fresh-Holdout Estimands and Uncertainty

For each dataset and backbone, let i index queries across its two seeds, $L_{ic} = -\log(\mathrm{clip}(p_{ic}(y_i), 10^{-12}, 1))$ denote candidate loss, and L_i^0 the development-selected fixed-reference loss. With $O_i = \mathbf{1}[\min_c L_{ic} < L_i^0]$, define

$$p_i = \frac{1}{16}\sum_{c=1}^{16} \mathbf{1}[L_{ic} < L_i^0], \qquad C_d^{\mathrm{rand}} = \frac{\sum_{i\in d} O_i p_i}{\sum_{i\in d} O_i}, \qquad C_d^{\pi} = \frac{\sum_{i\in d} O_i \mathbf{1}[L_{i,\pi(i)} < L_i^0]}{\sum_{i\in d} O_i}.$$

For each query, random conversion is calculated exactly by counting the slots that beat the reference and dividing by 16. Duplicate and reference slots are included; no Monte Carlo candidate draw is needed. Within each dataset, we combine success and bank-positive counts across the two seeds before computing policy and random conversion. The reported contrast is $100\ \mathrm{mean}_d(C_d^{\pi} - C_d^{\mathrm{rand}})$ in percentage points, weighting the ten datasets equally rather than pooling their query counts. These are the primary fresh-holdout estimates. The full-pool ratio-of-sums is a secondary descriptive summary, matching the aggregation form of Figure 6 but not the fresh-holdout primary estimand. For example, TabPFN P2 minus random is +8.89 pp under equal-dataset weighting versus +8.59 pp under the secondary ratio-of-sums summary.

Utility is $U_d^\pi = \frac{1}{2}\sum_{s=1}^{2} \mathrm{mean}_q(L_{dsq}^0 - L_{dsq,\pi(dsq)})$: it uses all queries, not only bank-positive queries. Post-bank gains use the same averaging with the specified comparator. Each backbone's estimate is the equally weighted mean over ten datasets. Pointwise percentile 95% intervals, without simultaneous or multiplicity adjustment across endpoints, use 10,000 paired dataset-bootstrap replicates, retaining paired methods and seeds within each resampled dataset. The dataset is the independent bootstrap unit. Secondary pooled-backbone results first average the two backbone-specific effects within each dataset, then resample datasets; they retain ten independent units. W/T/L counts positive, zero, and negative dataset effects.

The conversion comparison was planned in advance. The specific P2/P3 utility aggregation and bootstrap summary were specified after inference began, before local outcome evaluation, and are reported as supplementary analysis rather than fully prespecified confirmatory endpoints.

C Candidate Variation and Opportunity Composition

C.1 Candidate variation and dependence

Mean development Raw Oracle Headroom is 0.318082, with a descriptive 95% interval of [0.200231, 0.433146]. Mean absolute candidate-loss correlation is 0.620, and effective diversity is 2.30 for the nominal bank of 16 candidates. Absolute correlation averages the 120 pairwise absolute Pearson correlations. Effective diversity is $(\sum_r \lambda_r)^2 / \sum_r \lambda_r^2$, with negative correlation-matrix eigenvalues clipped to zero.

In the earlier four-dataset analysis, expanding the bank from 16 to 32 candidates adds 0.0192–0.0314 NLL of Raw Oracle Headroom, without a corresponding improvement from the fixed diagnostic policies. Because the larger bank contains every candidate in the smaller bank, its minimum loss cannot increase. The useful descriptive questions are therefore how much headroom is added, which queries account for it, and how it varies across datasets. This analysis is separate from the later PC B64 study and does not qualify a null procedure.

C.2 Exact Decomposition of Raw Oracle Headroom

A later descriptive analysis separates two sources of the observed gap on the 90 development unit–backbone loss surfaces. Each surface contains 256 queries and 16 candidate rules, with $K = 64$. For one unit and backbone, let $\bar{L}_c = Q^{-1}\sum_q L_{qc}$, $r_q = \frac{1}{3}\sum_{c\in\{0,8,9\}} L_{qc}$, and $\bar{r} = Q^{-1}\sum_q r_q$. Define

$$H = \bar{r} - \frac{1}{Q}\sum_q \min_c L_{qc},$$

$$G = \bar{r} - \min_c \bar{L}_c, \qquad S = \min_c \bar{L}_c - \frac{1}{Q}\sum_q \min_c L_{qc}. \tag{1}$$

Then $H = G + S$ is an algebraic identity. G is the fixed candidate-rule advantage over the reference; S is the observed per-query switching component. The best fixed candidate rule is chosen retrospectively on this surface, not the development-selected policy reference used in deployment comparisons. A rule can construct a different physical context for each query. Reference losses remain the mean of the three slot losses, not the loss of their averaged probabilities.

Table 8 averages five seeds within each dataset–shift cell, shifts within each dataset, and then datasets equally. These weights differ from the original 90-record pooled mean of 0.318082. Under the original record weighting, (H, G, S) are $(0.282521, 0.125271, 0.157251)$ for TabPFN and $(0.353643, 0.175707, 0.177936)$ for TabICL.

Mushroom's switching component is small despite substantial Raw Headroom, whereas Adult has a much larger switching component. Thus substantial headroom can arise from different candidate-loss organization. $S > 0$ describes additional hindsight opportunity from changing rules across queries; it is not noise-corrected interaction, predictable or recoverable structure, or deployable gain. Conversely, $S = 0$ would not imply that every form of non-additivity is absent: relative loss differences can vary while one rule remains best. This characterization requires no fitted additive model or significance test.

Table 8: Exact decomposition of development Raw Oracle Headroom (NLL). G is fixed candidate-rule advantage and S observed per-query switching opportunity. Seeds and shifts are averaged within dataset; the last row gives datasets equal weight. These are descriptive components of realized losses.

	TabPFN			TabICL		
Dataset	H	G	S	H	G	S
Adult	0.418035	0.127734	0.290301	0.513058	0.152105	0.360953
Dry Bean	0.346871	0.164257	0.182614	0.534501	0.346954	0.187547
Mushroom	0.144412	0.143461	0.000951	0.183222	0.181575	0.001648
Rice Cammeo/Osmancik	0.255538	0.091741	0.163797	0.282902	0.070789	0.212113
Spambase	0.234259	0.082396	0.151863	0.219162	0.074652	0.144510
Equal-dataset mean	0.279823	0.121918	0.157905	0.346569	0.165215	0.181354

C.3 Limited Cross-Seed Reproducibility Diagnostic

Does any query-specific advantage persist in the available historical repeats? Before calculating effects, we matched original source-row identities and labels, required exactly equal transformed query-feature vectors and compatible feature encodings, and checked candidate-rule identities, model versions, and complete banks. No matched query could occur in either support set. Rules could be instantiated on different support samples; this compares candidate rules, not identical physical contexts. At least two valid shared queries were required for a matching comparison. Eligibility did not depend on losses, winners, or effect direction.

Only Adult/class-prior, Dry Bean/density-mixture, and Spambase/local-support met these conditions. They supplied 9, 10, and 10 seed pairs, respectively: 29 pairs and 58 pair–backbone records. Other settings had different transformed features; Mushroom's category encodings also differed across seeds, even where some numerical vectors coincided. One Adult/class-prior pair had no shared query.

For a feasible pair A, B, let $\mathcal{Q}_{AB}$ contain its n shared queries and let $c_A(q) = \arg\min_c L_A(q, c)$, resolving exact ties by ascending slot. Let $f_A(c) = n^{-1} \sum_{q \in \mathcal{Q}_{AB}} \mathbf{1}[c_A(q) = c]$ be the source winner frequency. We calculate

$$M_{A \to B} = \frac{1}{n} \sum_{q \in \mathcal{Q}_{AB}} \left[\sum_c f_A(c) L_B(q, c) - L_B(q, c_A(q)) \right]. \tag{2}$$

The first term is the exact expected loss under uniform random reassignment of A's winner identities to B's queries, preserving their frequencies. The second retains the same-query correspondence. $M > 0$ means that this correspondence improves loss over that frequency-matched baseline. We also compute $M_{B \to A}$ and average the two directions within pair; Table 9 gives equal-pair descriptive summaries. No permutation test, confidence interval, or p-value is added.

Table 9: Limited cross-seed matching diagnostic. Mean M averages the two directions within pair and then weights seed pairs equally. Unique rows are counted within setting and shared across backbones. Positive counts are descriptive; no sign test is performed. Pair-level effects are not independent.

Setting	Backbone	Pairs	Unique rows	Mean M	$M > 0$ pairs
Adult/class-prior	TabPFN	9	34	0.061712	6/9
Adult/class-prior	TabICL	9	34	0.060046	5/9
Dry Bean/density-mixture	TabPFN	10	65	0.046818	6/10
Dry Bean/density-mixture	TabICL	10	65	0.035466	5/10
Spambase/local-support	TabPFN	10	164	0.082267	9/10
Spambase/local-support	TabICL	10	164	0.083717	9/10

All six setting–backbone mean effects are positive, but each contains negative pair-level effects. Signs are mixed for Adult and Dry Bean. Spambase/local-support has 9 of 10 positive symmetric effects

on each backbone, with means 0.082267 and 0.083717 for TabPFN and TabICL. This is partial, setting-specific evidence that query-identity-related advantages can persist; it neither explains all of S nor establishes general query-specific structure.

Coverage is sparse and dependent: there are 263 unique original query rows across the three settings, not independent samples for every direction or pair. Adult pairs contain 2–6 queries, Dry Bean 2–11, and Spambase 15–21. Pairs share seeds, some share queries, and the two backbones use the same queries. The 263 rows comprise 256 groups of identical original features. Support sets also overlap, and the same observed query label is used in both seeds. These comparisons are not independent replications or estimates under newly sampled query labels. Since the source winner uses the query's outcome, positive M does not validate pre-outcome selection or deployment value.

C.4 Historical N1/N2 Diagnostics

N1 expected headroom is 0.329474, with a reported observed-minus-null contrast of -0.011391. N2 expected headroom is 0.318151, with contrast -0.000069. N1 and N2 disagree in direction on 42 of 90 unit–backbone records.

N1 retains the candidate predictions and their dependence while drawing candidate-independent query labels from the averaged and renormalized reference-slot probability vectors. The implemented N2 diagnostic removes additive query and candidate effects from the loss matrix, resamples whole residual vectors across queries, and adds the effects back to reconstruct loss vectors. The empirical diagnostics use 2,000 draws and a plus-one upper-tail calculation. These describe the implemented empirical contrasts; similar aggregate means establish neither equivalence nor null acceptance.

For the implemented diagnostic on $L \in \mathbb{R}^{Q\times 16}$, $\mu = \overline{L}_{\cdot\cdot}$, $a_q = \overline{L}_{q\cdot} - \mu$, $b_c = \overline{L}_{\cdot c} - \mu$, and $r_{qc} = L_{qc} - \mu - a_q - b_c$; reconstruction uses $L^*_{qc} = \mu + a_q + b_c + r_{I_q c}$ with independent uniform donor indices I_q, including the reference coordinates, without scale estimation, residual standardization, or clipping. The common query effect cancels from reference-minus-minimum headroom, giving $H^*(q) = H^{\mathrm{raw}}(I_q)$ and hence $\mathbb{E}[H^*(q) \mid L] = Q^{-1}\sum_i H^{\mathrm{raw}}(i)$: each resampled headroom equals its donor row's headroom. The mean-headroom contrast is therefore non-identifying under this construction: a near-zero finite-draw contrast cannot test whether residual structure is absent.

The specified N2 model is $L_{uqc} = \mu_u + a_{uq} + b_{uc} + s_{uq}\epsilon_{uqc}$. The implemented diagnostic resamples each unstandardized residual vector as a whole, retaining within-vector dependence. It estimates additive effects on the same query matrix and does not use cross-fitting or separate training folds. N1 reference slots use seeds 101, 211, and 307. The plus-one Monte Carlo calculation follows the cited convention (Phipson & Smyth, 2010); the planned cross-fitting and the small-cluster stress are motivated by their cited methodological sources (Chernozhukov et al., 2018; Cameron et al., 2008; MacKinnon & Webb, 2017).

C.5 Historical Qualification Limitations

C.5.1 Score-Level Simulation

The implemented score-level simulations reveal a conservative testing regime. We evaluated N1 and N2 at query counts of 256 and 512. Type-I error and coverage met the implemented criteria, and bias met each endpoint's criterion. Across all 208 required power evaluations, estimates were 0.0240–0.0561, below the prespecified criterion. Fidelity was incomplete, and the small-cluster checks did not meet qualification criteria. We distinguish primary scenarios with 10,000 repetitions from diagnostic runs. The four-cluster stress test is a simulation, not an additional collection of real datasets; its power also remained low.

These results characterize the implemented score-level procedure, not the intended empirical estimator end to end. Low null rejection rates indicate conservatism, but we cannot separate its contribution to low power from other simulation limitations. The findings justify neither null acceptance nor a claim that residual structure is intrinsically difficult to detect.

C.5.2 Differences Between the Protocol and Implementation

Reviewing the implementation against the protocol reveals three limits to this interpretation. First, the protocol specification assigns the designated tail distributions to both U and V and places the specified heteroskedastic scaling on V. The implementation applies the designated tail and scaling to the between-cluster U term while using a standard-normal within-cluster V term. Second, the primary simulation generates scalar endpoint cluster scores rather than complete query-by-candidate losses followed by the empirical N1/N2 estimation procedure. Diagnostics of candidate-vector fidelity do not validate the complete estimation procedure. Third, the reported p-value grid lists values allowed by the arithmetic calculation; it does not show that the test can attain every listed value. With symmetric Webb weights and an absolute two-sided statistic, counts occur in pairs in exact arithmetic, restricting which values are attainable.

These implementation differences limit transfer to the intended estimator; the non-identifying N2 contrast derived above is unchanged. A corrected end-to-end study was considered, but some joint simulation scenarios remained numerically unresolved, and the protocol did not uniquely specify how to evaluate the N2 population target $\mathbb{E}[(T - J)/D]$, where J is the null expectation of T and T, D are defined in Appendix B. Resolving these gaps would require new choices of joint data-generating process, truth evaluation, and estimator mapping, rather than a uniquely specified correction.

We therefore did not run a corrected end-to-end study or treat the missing qualification as evidence for or against residual structure.

Table 10: Prespecified criteria for evaluating the N1/N2 procedures. Primary scenarios use 10,000 simulation repetitions; the 2,000-repetition runs are diagnostic.

Criterion	Operational statistic	Pre-specified criterion	Scenario	Purpose
Type-I	Rejection rate under candidate-independent null draws	Point ≤ 0.055; one-sided 95% Monte Carlo upper bound ≤ 0.060	Four-cluster primary scenarios; 10,000 repetitions	Controls false positive behavior at nominal $\alpha = 0.05$.
Coverage	Coverage of the pre-specified interval	Point ≥ 0.94; one-sided 95% lower bound ≥ 0.93; row floor 0.93	Four-cluster primary scenarios; 10,000 repetitions	Checks interval calibration.
Bias	Absolute mean error for pre-specified endpoints	Mean NLL gain ≤ 0.001; N1/N2 adjusted headroom ≤ 0.001; structure fraction ≤ 0.005; CVaR90 harm ≤ 0.004	Fixed null and tail scenarios	Checks centering of the estimand-specific statistic.
Power	Rejection probability at the pre-specified stress effect	Point and Wilson 95% lower bound ≥ 0.80 at standardized effect 0.60	Primary stress rows; lower 0.10/0.30 rows are diagnostic	Checks ability to detect the pre-specified alternative.
Fidelity	Standardized mean/SD error, correlation-matrix MAE, q90/q95 error, effective-diversity relative error, and missingness error	Mean/SD 0.10; corr. MAE 0.10; q90/q95 0.15; effective diversity 0.20; missingness 0.02	Simulated dependence -preserving rows	Checks whether the simulated candidate surface matches the target dependence structure.

Continued on next page

Table 10 continued

Criterion	Operational statistic	Pre-specified criterion	Scenario	Purpose
Small -cluster	Restricted studentized wild cluster bootstrap-t Type-I error, coverage, and power	Webb weights; four clusters; same Type-I error, coverage, and power criteria as above	$G = 4$ clusters; NORMAL/T5/T3; heteroskedastic ratios 1.0/1.5/2.0; $G = 3, 8$ diagnostic	Checks the evaluated small-cluster inference regime; Not met is a method qualification result, not a scientific null.

Table 11: Qualification results for the implemented score-level simulations of N1 and N2 at each query count. A procedure must meet every criterion to qualify. These simulations do not validate the complete empirical estimators.

Null/Q	Type-I	Coverage	Bias	Power	Fidelity	Small cluster	Qualified
N1 / Q256	Met	Met	Met	Not met	Not met	Not met	No
N1 / Q512	Met	Met	Met	Not met	Not met	Not met	No
N2 / Q256	Met	Met	Met	Not met	Not met	Not met	No
N2 / Q512	Met	Met	Met	Not met	Not met	Not met	No

Table 12: Numerical results for the four null-by-query-count combinations. Ranges summarize the evaluated simulation scenarios. Maximum bias and the endpoint-specific breakdown use the three null scenarios, pooling both K values and models within each null/query-count combination. Each endpoint is assessed against its own bias threshold.

Statistic	N1/Q256	N2/Q256	N1/Q512	N2/Q512
Type-I error point range	0.0017–0.0035	0.0012–0.0039	0.0017–0.0035	0.0017–0.0037
Coverage point range	0.9629–0.9708	0.9634–0.9704	0.9646–0.9715	0.9625–0.9713
Maximum absolute bias	0.002788	0.003212	0.001711	0.002088
Power at pre-specified stress effect	0.0297–0.0523	0.0247–0.0434	0.0291–0.0561	0.0240–0.0421
Wilson 95% lower-bound range	0.0270–0.0488	0.0223–0.0402	0.0265–0.0524	0.0216–0.0389
Fidelity rows meeting criteria	11/12	11/12	11/12	11/12
Worst fidelity SD absolute error	1.151643	0.241372	1.151643	0.241372
Maximum absolute bias by endpoint (null scenarios; threshold in parentheses)				
Mean NLL gain (≤ 0.001)	0.000486	0.000816	0.000538	0.000507
Adjusted headroom (≤ 0.001)	0.000599	0.000552	0.000584	0.000455
Structure fraction (≤ 0.005)	0.002788	0.003212	0.001628	0.002088
CVaR90 harm (≤ 0.004)	0.000731	0.002090	0.001711	0.001208

The separate $G = 4$ small-cluster stress has Type-I ranges 0.0333–0.0489, coverage 0.9511–0.9667, and power at 0.60 of 0.1241–0.1976, with none of the nine scenarios meeting all prespecified criteria. These values come from the fixed qualification analysis and are not additional real-data clusters.

C.6 Older Four-Dataset Raw Oracle Sensitivity

C.6.1 Design

The design fixes the per-context budget at $K = 64$, the query prefix at Q256, and uses a distinct four-dataset supplementary set: Banknote Authentication (Lohweg, 2012), HTRU2 (Lyon, 2015), Landsat (Srinivasan, 1993), and Student Dropout (Realinho et al., 2021). It evaluates TabPFN v2

and TabICL v1 under 15% query-side MCAR and nuisance shifts, with five fixed seeds per cell. Candidate-bank size is $B \in \{4, 8, 16, 32\}$. For each group, Bank-32 is constructed once. Bank-4, Bank-8, and Bank-16 are prefixes of the same prespecified candidate ordering. Candidate order was not changed using outcomes. Figure 4 shows the four dataset trajectories and pooled means, Figure 5 shows the dataset-level $B32 - B16$ increments, and Table 13 reports the compact pooled comparison. The figures give the four datasets equal weight. Results for alternative orderings are descriptive only. This nested-bank comparison does not hold candidate composition fixed while intervening on B, and it does not establish robustness to another K, learned bank, optimized bank, or qualitatively different candidate family.

C.6.2 Raw Oracle Scaling

Since every larger bank includes the smaller one, the hindsight minimum cannot get worse. We therefore examine the size of the improvement and how it varies across datasets, rather than treating monotonicity itself as a finding. The equal-weight pooled trajectories for $B = (4, 8, 16, 32)$ are $(0.162, 0.220, 0.282, 0.313)$ for TabPFN under 15% query-side MCAR, $(0.110, 0.152, 0.191, 0.211)$ for TabPFN under the nuisance shift, $(0.143, 0.200, 0.255, 0.284)$ for TabICL under 15% query-side MCAR, and $(0.132, 0.181, 0.225, 0.244)$ for TabICL under the nuisance shift. The corresponding $B32 - B16$ Raw Oracle increments are 0.031, 0.021, 0.028, and 0.019.

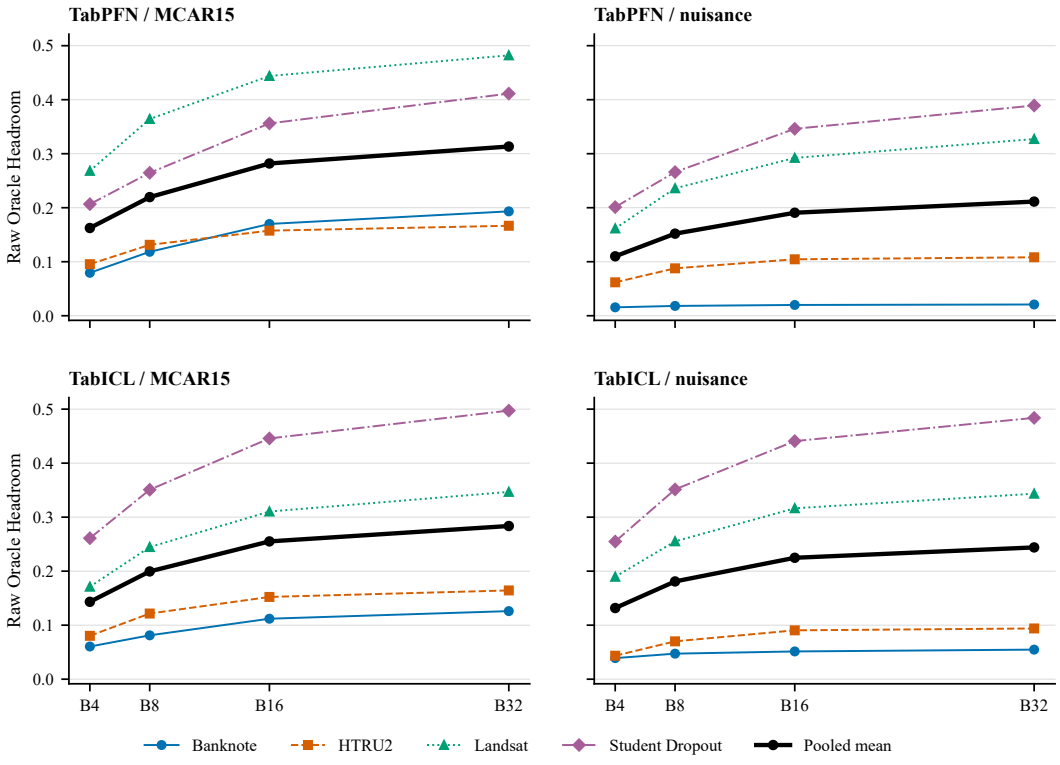


Figure 4: Raw Oracle Headroom as the bank grows from 4 to 32 candidates, with K = 64. Colored lines show the four datasets, and the pooled line gives their equal-weight mean. MCAR15 masks 15% of query feature entries. Because each larger bank contains the smaller bank, headroom cannot decrease. The figure describes the size and dataset variation of the increases; it does not show confidence intervals or a significance threshold.

C.6.3 Redundancy and Concentration

The added bank members were highly redundant. At B32, mean absolute candidate-loss correlation ranged from 0.787 to 0.831 across the four backbone-by-shift settings, while effective diversity ranged from 1.443 to 1.589. The B16-to-B32 effective-diversity increments were only 0.039–0.065.

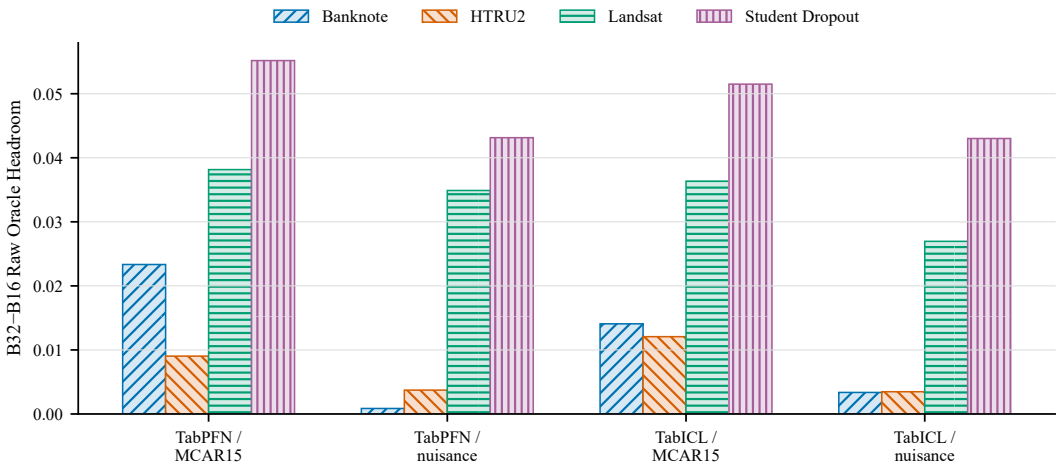


Figure 5: Increase in Raw Oracle Headroom from 16 to 32 candidates for each dataset and model–shift setting. The bars show how the increase differs across datasets; they are descriptive contributions within a setting, not additional independent replications.

Across the 80 groups, 56.59% of $B32 - B16$ query increments were zero. Within each group, we rank queries by increment and divide the sum for the top 1%, 5%, or 10% by that group's total increment. Averaging these shares equally across groups gives 31.47%, 63.50%, and 79.39%, respectively. Counts are rounded up, so the top 1% includes three of the 256 queries per group. Separately, deleting the single maximum-increment query in each group reduces its total by that query's share; the equal-group mean reduction is 14.99%. Fixed 99% winsorization gives a mean reduction of 12.72% under the same group-wise normalization. These diagnostics indicate that one isolated outlier does not account for the increment. No single new slot dominated the selections: the largest single new-slot share was 10.10–14.40%, the top four accounted for 36.70–42.20%, and all new slots together supplied the hindsight-selected context on 40.00–46.50% of queries. These are descriptive checks on the existing outputs, not new inferential tests.

The increment was dataset-dependent. Student Dropout plus Landsat accounted for 74.25–77.07% of the $B32 - B16$ increment under 15% query-side MCAR and 91.10–94.45% under the nuisance shift across the two backbones. The smaller increases for Banknote and HTRU2 show that the larger gains are not shared uniformly by all four datasets.

C.6.4 Deployment Conversion

Table 13 compares the pooled B16 and B32 Raw Oracle values with opportunity-rate, redundancy, conversion, and paired-gain changes. P2 conversion decreases by 0.0045–0.0181 and P3 conversion by 0.0044–0.0175; paired-gain changes range from -0.00042 to $+0.00001$. For P2/P3, the pooled $h > 0.05$ harm rate and P95/CVaR90 harm are unchanged or increase; P1 has mixed changes. Fallback rates are unchanged for all three policies. The larger bank therefore offers more improvement to the hindsight Oracle without a corresponding gain for these fixed policies. This finding concerns the evaluated policy panel; it does not predict how a different selector would perform or validate a null model.

Table 13: Candidate-bank-size sensitivity under fixed $K = 64$. Entries pool the four supplementary datasets with equal weights for the fixed prospective candidate ordering. All deltas are B32 minus B16; P2 and P3 are shown separately so that conversion and paired-gain changes are not conflated. The lower panel gives descriptive ranges across the four backbone-by-shift settings, averaging the reported statistics over seeds within each cell and then equally over datasets; these are not confidence intervals or pooled-query tail estimates. Positive harm changes mean greater harm; rates are proportions.

Setting	Policy	B16 Raw	B32 Raw	ΔRaw	ΔOpp.	ΔEff. Div.	ΔConv.	ΔGain
TabPFN / MCAR15	P2	0.28188	0.31330	+0.03143	+0.01152	+0.03949	−0.00634	−0.00028
	P3	0.28188	0.31330	+0.03143	+0.01152	+0.03949	−0.00604	−0.00028
TabPFN / nuisance	P2	0.19069	0.21135	+0.02066	+0.00703	+0.04743	−0.00453	−0.00004
	P3	0.19069	0.21135	+0.02066	+0.00703	+0.04743	−0.00436	−0.00011
TabICL / MCAR15	P2	0.25510	0.28360	+0.02850	+0.02773	+0.04825	−0.01807	−0.00042
	P3	0.25510	0.28360	+0.02850	+0.02773	+0.04825	−0.01751	−0.00042
TabICL / nuisance	P2	0.22477	0.24397	+0.01920	+0.02520	+0.06458	−0.01810	+0.00000
	P3	0.22477	0.24397	+0.01920	+0.02520	+0.06458	−0.01578	+0.00001

The nested construction makes the raw best-of-prefix trajectory non-decreasing by definition; the sensitivity concerns the size of the incremental variation, its redundancy, and whether the fixed policies recover it. No new null analysis, p-value, or uncertainty estimate is introduced here.

Policy	$\Delta \Pr(h > 0.05)$	ΔP95 harm	ΔCVaR90 harm	ΔFallback
P1	$[-0.000586, +0.000195]$	$[+0.000000, +0.001132]$	$[-0.000018, +0.000761]$	0
P2	$[+0.000000, +0.000781]$	$[+0.000000, +0.001023]$	$[+0.000000, +0.003039]$	0
P3	$[+0.000195, +0.000781]$	$[+0.000000, +0.001457]$	$[+0.000000, +0.003059]$	0

D Single-Call Policy Evaluation and Diagnostics

P1 chooses contexts by proximity to the query. P2 chooses by geometric class coverage. P3 uses P2's choice unless its trigger calls for the development-selected fixed reference instead. P3 still returns a prediction; it does not abstain. These decisions cannot use candidate predictions, query labels, realized losses, or the hindsight-best candidate.

D.1 Opportunity Prevalence

At least one candidate beats the fixed reference for 19,268 of 20,480 exploratory observations (94.08%) and 9,922 of 10,240 reserve observations (96.89%).

D.2 Policy Conversion

Among these observations, P1, P2, and P3 select a context that beats the reference 36.25%, 55.82%, and 52.38% of the time in exploration, and 49.83%, 63.82%, and 60.80% in reserve.

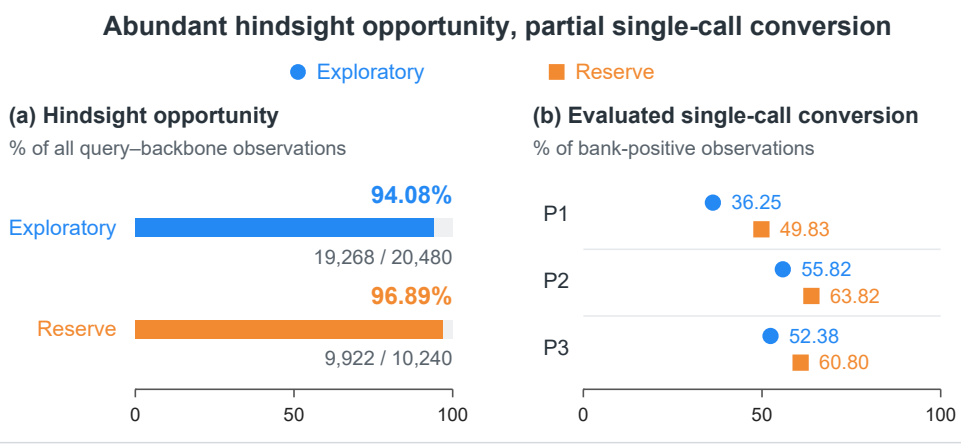


Figure 6: Available opportunities and the fraction selected by P1–P3 in the exploratory and reserve evaluations. (a) Fraction of all query–model observations for which at least one candidate beats the fixed reference. (b) Among those observations, the fraction where each policy selects a context that beats the reference. These are success rates, not fractions of Oracle NLL headroom recovered or measures of final deployment value. In panel (b), blue circles show exploratory results and amber squares show reserve results.

Table 14: Expected conditional conversion under uniform random selection, calculated after the original evaluations from the saved Bank-16 predictions and labels. Conversion counts only queries for which at least one candidate beats the fixed reference. Pooled rows combine success and opportunity counts before taking their ratio, as in Figure 6. Policy-minus-random differences are in percentage points.

Evidence set	Backbone	Random conv.	P1−random	P2−random	P3−random
Exploratory	TabPFN	44.96%	−32.62	+6.00	+2.91
Exploratory	TabICL	51.23%	+8.56	+9.37	+5.58
Exploratory pooled	Both	48.12%	−11.87	+7.70	+4.26
Reserve	TabPFN	51.57%	−0.08	+16.69	+13.53
Reserve	TabICL	50.50%	−2.38	+8.74	+5.87
Reserve pooled	Both	51.04%	−1.21	+12.78	+9.76

Table 14 gives the exact expected conversion rate for uniform random choice among the 16 candidate slots: $\sum_i O_i p_i / \sum_i O_i$, where $O_i = \mathbf{1}[\min_c L_{ic} < L_i^0]$ and $p_i = 16^{-1} \sum_{c=1}^{16} \mathbf{1}[L_{ic} < L_i^0]$. All 16 slots are retained, including duplicate identities and reference slots. This measures the probability of beating the fixed reference, rather than the expected NLL reported for uniform-random selection in Appendix F. We reconstructed reserve losses from the saved probabilities and original labels, retaining the original class mapping, query order, and NLL clipping. This reproduced the opportunity and policy-conversion results. The random comparison is a later descriptive analysis of these outputs, not a new confirmatory test. P2 and P3 exceed random conversion in all eight reserve dataset–backbone cells, with margins of 6.33–20.84 and 4.76–16.77 percentage points, respectively; P1 is above random in four cells and below it in four. In exploration, P2 exceeds random conversion in seven of the eight dataset–model comparisons, and P3 in six. P1 is below random in all four TabPFN comparisons and above random in all four TabICL comparisons.

D.3 Ex-Post Policy Union

At least one of P1–P3 selects a context that beats the reference for 12,051 of the 19,268 exploratory opportunities (62.54%) and 7,253 of the 9,922 reserve opportunities (73.10%). P1 and P2 succeed on some of the same queries, but each also succeeds where the other does not. P3 cannot add a

success beyond P2: it either follows P2 or returns the reference. These union rates are calculated after outcomes are known and do not describe a deployable rule for choosing among the policies.

D.4 Mean Policy Utility

All three policies have positive average gains on both backbones in both the original exploratory and reserve evaluations. Some individual datasets nevertheless have negative gains.

D.5 LODO and Influence Sensitivity

Across the four leave-one-dataset-out evaluations, exploratory TabPFN gains relative to the fixed reference range from -0.019652 to $+0.038928$ for P2 and from -0.021168 to $+0.031356$ for P3. Both policies have negative gains when Bank Marketing or Online Shoppers is omitted and positive gains when Covertype or Wine Quality is omitted. These gains compare each policy with the fixed reference. Table 25 instead compares P3 with P2 under LODO. Separately, the exploratory prespecified extreme-query deletion changes TabPFN P3 gain from +0.0040 to -0.0092 (Table 27). The K64-only development reference-ranking sensitivity does not replace the primary fixed reference; reserve alternate-reference utility was not included in the original sensitivity analysis.

D.6 Rate-Matched Fallback

P3 returns the fixed reference on flagged queries, making gain zero there. Compared with P2, this cannot increase the fraction with $h > 0$ or $h > 0.05$. In the original exploratory and reserve evaluations, the rate-matched comparison preserves the observed fallback count within each cell of the relevant evidence stratum and assigns that number of fallbacks uniformly, using 10,000 conditional draws. P3 has higher ordinary and h > 0.05 harm than the random-fallback mean in all four stratum-by-backbone combinations; tail measures do not consistently favor either rule. The Monte Carlo ranges describe changes under random fallback assignment within these observations, not uncertainty over new datasets. We also report whether flags identify later harm and how often P1 and P2 choose different candidates. Disagreement alone does not indicate which policy is better.

D.7 Opportunity-Threshold Sensitivity

Existing-loss sensitivity evaluates $\Pr(L^0 - \min_c L_c > \varepsilon)$ for $\varepsilon \in \{0, 10^{-4}, 10^{-3}, 0.01, 0.05\}$. At 0.01, prevalence is 92.47% in exploration and 80.50% in reserve; at 0.05 it is 82.69% and 59.34%. These later checks describe the size of available loss reductions. The primary definition still counts any strictly positive reduction; the thresholds do not establish practical importance for a particular application.

D.8 Policy Definitions and Fixed-Reference Sensitivity

For P1, we find the distance from the query to the nearest support example in each candidate context and select the candidate with the smallest distance. Ties are broken first by the mean query-to-support distance, then by candidate identity.

For P2, we find the nearest support example from each source class within a candidate context. We score that context by the largest of these class-specific distances and choose the context with the smallest score. Ties are broken by candidate identity. Numeric distances use source-fit standardization, and categorical coordinates contribute mismatch indicators.

P3 normally uses P2's context. It falls back to the development-selected fixed reference in Bank-16 when the context is missing a source class, its coverage distance exceeds the source-side leave-one-out 0.95 quantile, or an integrity check fails. The latter includes a missing or non-finite coverage decision, or a context without exactly 64 unique support row identifiers in the required positions. The policy cannot inspect query labels, candidate predictions or losses, the hindsight-best candidate, or other members of the query batch. It always makes a prediction. The implementation name SAFE_COVERAGE_FALLBACK does not imply that the rule has a validated safety guarantee.

For global fixed-policy selection, development seeds 11 and 29 were used with equal unit weight; validation and evaluation outcomes were not used for selection or tie-breaking. We select the identity

with the lowest mean query NLL, weighting development units equally. Ties are broken by lower deployment TFM-call cost and then lexicographic policy identity, as specified in advance. Table 15 shows the top-five development scores for each backbone. TabICL has six eligible units from three compatible $K = 64$ development cells and two seeds; TabPFN has 24 from the corresponding cells across $K \in \{32, 64, 128, 256\}$ and the evaluated bank variants. The eligible units are those compatible with each model in the development grid; eligibility is not chosen from evaluation outcomes. Reference slots 0, 8, and 9 use seeds 101, 211, and 307; in the policy and reserve Raw Oracle estimand, the reference-slot mean is the arithmetic mean of their three per-slot query NLLs. This is a within-stratum reference-family average, not a cross-stratum comparison.

The development-only ranking was recomputed after restricting eligibility to $K = 64$, with seeds 11 and 29, equal-unit weighting, candidate identities, and tie-breaks unchanged. TabICL retains `RANDOM_SLOT_6`; TabPFN changes from `STANDARDIZED_KNN`, selected by the original multi-K development procedure, to the $K = 64$ sensitivity winner `RANDOM_SLOT_6`. The prespecified primary comparator remains unchanged. Table 16 gives the top-five rankings and gaps; the available exploratory alternate-reference utility is reported below the table. Reserve alternate-reference utility was not included in the original sensitivity analysis.

Table 15: Development-only top-five fixed-reference selection. Lower mean query NLL is better.

Backbone	Rank	Policy identity	Mean query NLL	Units	Selected
TabICL v1	1	`RANDOM_SLOT_6`	0.514967	6	Yes
	2	`REFERENCE_SLOT_0`	0.523848	6	
	3	`LOCAL_FARTHEST_FIRST`	0.524419	6	
	4	`STANDARDIZED_KNN`	0.529055	6	
	5	`RANDOM_SLOT_5`	0.530410	6	
TabPFN v2	1	`STANDARDIZED_KNN`	0.493360	24	Yes
	2	`RANDOM_SLOT_0`	0.496158	24	
	3	`LOCAL_FARTHEST_FIRST`	0.496437	24	
	4	`RANDOM_SLOT_5`	0.498765	24	
	5	`RANDOM_SLOT_6`	0.503996	24	

Table 16: $K = 64$-only fixed-reference ranking sensitivity using the original development evaluation. Lower mean query NLL is better; the primary comparator is not redefined.

Backbone	$K = 64$ rank	Identity	Mean NLL	Primary identity?
TabICL	1	`RANDOM_SLOT_6`	0.514967	Yes
	2	`REFERENCE_SLOT_0`	0.523848	No
	3	`LOCAL_FARTHEST_FIRST`	0.524419	No
	4	`STANDARDIZED_KNN`	0.529055	No
	5	`RANDOM_SLOT_5`	0.530410	No
TabPFN	1	`RANDOM_SLOT_6`	0.489821	No
	2	`LOCAL_FARTHEST_FIRST`	0.495317	No
	3	`RANDOM_SLOT_5`	0.495605	No
	4	`SINGLETON_MMD`	0.496226	No
	5	`STANDARDIZED_KNN`	0.507470	Yes

Ranking gaps: the $K = 64$ NLL gaps between the best and second-ranked candidates are 0.008881 for TabICL and 0.005497 for TabPFN; the corresponding gaps under the original selection across multiple K values are 0.008881 and 0.002798. Exploratory alternate-reference utility for TabPFN `RANDOM_SLOT_6` is P1 = 0.007461, P2 = 0.007084, and P3 = 0.002177. Under the same exploratory aggregation, the primary TabPFN reference has 0.001841 higher NLL than this $K = 64$ sensitivity winner. Reserve alternate-reference utility was not included in the original sensitivity analysis.

D.9 Complete Policy and Influence Results

Table 17 summarizes mean policy utility and the fallback comparisons.

Table 17: Fixed-reference utility and fallback comparisons. Positive gain favors the policy over the development-selected fixed reference; negative harm deltas favor P3. In Panels B and C, positive ΔGain favors P3. In Panel C, positive harm deltas mean that P3 is worse than rate-matched random assignment. Reserve values retain equal-dataset aggregation over four independent datasets.

Panel A: Mean gain relative to the development-selected fixed reference

Evidence set	Backbone	P1 vs ref.	P2 vs ref.	P3 vs ref.	P3−P2
Exploratory	TabICL	+0.0405	+0.0232	+0.0201	−0.0031
Exploratory	TabPFN	+0.0093	+0.0089	+0.0040	−0.0049
Reserve	TabICL	+0.0003	+0.0023	+0.0028	+0.0006
Reserve	TabPFN	+0.0015	+0.0104	+0.0100	−0.0004

Panel B: Descriptive P3−P2 effect

Evidence set	Backbone	Fallback	ΔHarm	$\Delta(h > 0.05)$	ΔP95	ΔCVaR90	ΔGain
Exploratory	TabICL	5.21%	−0.0161	−0.0147	−0.0189	−0.0260	−0.0031
Exploratory	TabPFN	5.21%	−0.0232	−0.0149	−0.0145	−0.0211	−0.0049
Reserve	TabICL	4.30%	−0.0156	−0.0059	−0.0135	−0.0147	+0.0006
Reserve	TabPFN	4.30%	−0.0119	−0.0051	−0.0093	−0.0113	−0.0004

Panel C: P3 minus rate-matched random-fallback mean

Evidence set	Backbone	ΔHarm	$\Delta(h > 0.05)$	ΔP95	ΔCVaR90	ΔGain
Exploratory	TabICL	+0.0045	+0.0025	+0.0043	+0.0013	−0.0023
Exploratory	TabPFN	+0.0043	+0.0061	−0.0027	−0.0063	−0.0051
Reserve	TabICL	+0.0025	+0.0034	+0.0019	−0.0008	+0.0007
Reserve	TabPFN	+0.0019	+0.0009	−0.0013	−0.0024	+0.0002

Panel C uses 10,000 conditional random assignments per evidence-set/backbone combination, with P3's observed fallback count held fixed within each cell of the relevant evidence stratum. These randomization results condition on the observed fallback counts within each cell. A tail probability from this distribution describes random fallback assignment on these observations, not performance on new datasets.

Table 18: Exploratory opportunity and conversion by dataset and backbone. Each cell has $n = 2{,}560$; conversion, union, and missed percentages condition on the displayed bank-positive count.

Dataset	Model	Bank n (%)	P1 conv.	P2 conv.	P3 conv.	Union	Missed
Bank Marketing	TabICL	2507 (97.93)	63.06	74.31	67.49	79.06	20.94
Bank Marketing	TabPFN	2364 (92.34)	19.63	50.72	44.88	57.99	42.01
Covertype	TabICL	2352 (91.88)	59.61	52.34	51.06	66.37	33.63
Covertype	TabPFN	2332 (91.09)	0.04	45.20	43.65	45.20	54.80
Online Shoppers	TabICL	2465 (96.29)	55.54	57.08	54.36	66.41	33.59
Online Shoppers	TabPFN	2457 (95.98)	28.90	53.81	52.22	63.09	36.91
Wine Quality	TabICL	2386 (93.20)	60.90	57.96	53.77	66.89	33.11
Wine Quality	TabPFN	2405 (93.95)	0.21	53.89	50.48	54.05	45.95

Table 19: Reserve opportunity and conversion by dataset and backbone. Each cell has $n = 1{,}280$; conditional denominators are the displayed bank-positive counts.

Dataset	Model	Bank n (%)	P1 conv.	P2 conv.	P3 conv.	Union	Missed
Breast Cancer	TabICL	1209 (94.45)	29.45	49.30	47.73	51.94	48.06
Breast Cancer	TabPFN	1239 (96.80)	50.77	76.51	73.45	86.76	13.24
Phishing	TabICL	1236 (96.56)	57.36	64.00	62.62	68.45	31.55
Phishing	TabPFN	1265 (98.83)	53.44	63.56	62.37	78.50	21.50
Heart Disease	TabICL	1173 (91.64)	55.24	63.51	59.25	68.20	31.80
Heart Disease	TabPFN	1266 (98.91)	53.55	71.41	66.35	84.12	15.88
Raisin	TabICL	1266 (98.91)	50.32	60.11	55.85	68.17	31.83
Raisin	TabPFN	1268 (99.06)	48.19	61.75	58.44	77.52	22.48

Table 20: Exploratory P3−P2 harm endpoints and utility by dataset and backbone. Each cell contains 2,560 query observations. The fallback rate is displayed as a percentage; negative harm deltas favor P3, while gain remains a separate utility quantity.

Dataset	Model	Fallback rate	ΔHarm	$\Delta(h > 0.05)$	ΔP95	ΔCVaR90	ΔGain
Bank Marketing	TabICL	8.09%	-0.01406	-0.01094	-0.01372	-0.02290	-0.00720
Bank Marketing	TabPFN	8.09%	-0.02695	-0.01328	-0.00590	-0.01229	-0.01089
Covertype	TabICL	3.13%	-0.01953	-0.01836	-0.00997	-0.00837	-0.00022
Covertype	TabPFN	3.13%	-0.01719	-0.01484	-0.01483	-0.01805	+0.00309
Online Shoppers	TabICL	3.56%	-0.00938	-0.00937	-0.03254	-0.04793	+0.00101
Online Shoppers	TabPFN	3.56%	-0.02031	-0.00664	-0.01241	-0.02566	-0.01508
Wine Quality	TabICL	6.06%	-0.02148	-0.02031	-0.01955	-0.02481	-0.00614
Wine Quality	TabPFN	6.06%	-0.02852	-0.02500	-0.02475	-0.02843	+0.00325

Table 21: Reserve P3−P2 harm endpoints and utility by dataset and backbone. Each cell contains 1,280 query observations. The fallback rate is displayed as a percentage; the four datasets, not the eight rows, are independent clusters.

Dataset	Model	Fallback rate	ΔHarm	$\Delta(h > 0.05)$	ΔP95	ΔCVaR90	ΔGain
Breast Cancer	TabICL	4.06%	-0.02578	-0.00234	-0.00447	-0.00763	-0.00024
Breast Cancer	TabPFN	4.06%	-0.01094	-0.00234	-0.00619	-0.00288	-0.00045
Phishing	TabICL	2.19%	-0.00859	-0.00469	-0.01427	-0.01240	+0.00045
Phishing	TabPFN	2.19%	-0.01016	-0.00313	-0.00437	-0.00853	+0.00022
Heart Disease	TabICL	6.41%	-0.02500	-0.01484	-0.03286	-0.02859	+0.00186
Heart Disease	TabPFN	6.41%	-0.01406	-0.01016	-0.02657	-0.03363	+0.00069
Raisin	TabICL	4.53%	-0.00313	-0.00156	-0.00229	-0.01000	+0.00014
Raisin	TabPFN	4.53%	-0.01250	-0.00469	0.00000	0.00000	-0.00196

Table 22: How well the P3 fallback flag identifies later harm. For each endpoint, sensitivity is the fraction of positive observations flagged, and specificity is the fraction of negative observations left unflagged. Reserve rows pool the specified reserve observations. These results do not validate the flag as a risk detector.

Evidence set	Scope	Endpoint	Population	Positive	Negative	Sensitivity	Specificity
Exploratory	TabICL	Harm	10,240	4,345	5,895	3.80%	93.76%
Exploratory	TabICL	$h > 0.05$	10,240	3,730	6,510	4.05%	94.13%
Exploratory	TabPFN	Harm	10,240	5,322	4,918	4.47%	94.00%
Exploratory	TabPFN	$h > 0.05$	10,240	4,193	6,047	3.65%	93.72%
Reserve	Pooled	Harm	10,240	3,908	6,332	3.61%	95.28%
Reserve	Pooled	$h > 0.05$	10,240	1,762	8,478	3.18%	95.47%

Table 23: Later outcomes among flagged and unflagged observations. Lower harm rates in the flagged group mean that the trigger does not concentrate harm in that group. Rates pool both models within each evaluation set; the exploratory comparisons are descriptive.

Evidence set	Later outcome	Flagged rate	Unflagged rate	Rate difference	Rate ratio
Exploratory	Bank opportunity	93.25%	94.13%	-0.88 pp	0.991
Exploratory	P2 harm	37.80%	47.72%	-9.91 pp	0.792
Exploratory	P2 $h > 0.05$ harm	28.52%	39.24%	-10.73 pp	0.727
Reserve	P2 harm	32.05%	38.44%	-6.39 pp	0.834
Reserve	P2 $h > 0.05$ harm	12.73%	17.41%	-4.68 pp	0.731

Table 24: How often P1 and P2 select different candidates. The selections do not depend on the backbone, so each query decision is counted once. Disagreement is available before the label is known, but does not tell us which policy will have lower loss.

Evidence set	Dataset	Population	Disagree	Agree	Disagree (%)
Exploratory	Bank Marketing	2,560	2,477	83	96.76
Exploratory	Covertype	2,560	2,560	0	100.00
Exploratory	Online Shoppers	2,560	2,094	466	81.80
Exploratory	Wine Quality	2,560	2,558	2	99.92
Reserve	Breast Cancer	1,280	1,280	0	100.00
Reserve	Phishing	1,280	1,273	7	99.45
Reserve	Heart Disease	1,280	1,279	1	99.92
Reserve	Raisin	1,280	1,275	5	99.61

Overall disagreement is 9,689/10,240 in the exploratory analysis and 5,107/5,120 in the reserve analysis. Backbone-specific outcome scoring remains separate elsewhere.

Table 25: Exploratory leave-one-dataset-out P3−P2 contrasts. Negative harm deltas favor P3.

Model	Omitted dataset	ΔHarm	$\Delta(h > 0.05)$	ΔP95	ΔCVaR90	ΔGain
TabICL	Bank Marketing	-0.01680	-0.01602	-0.02069	-0.02703	-0.00178
TabICL	Covertype	-0.01497	-0.01354	-0.02194	-0.03188	-0.00411
TabICL	Online Shoppers	-0.01836	-0.01654	-0.01441	-0.01869	-0.00452
TabICL	Wine Quality	-0.01432	-0.01289	-0.01874	-0.02640	-0.00214
TabPFN	Bank Marketing	-0.02201	-0.01549	-0.01733	-0.02405	-0.00291
TabPFN	Covertype	-0.02526	-0.01497	-0.01435	-0.02213	-0.00757
TabPFN	Online Shoppers	-0.02422	-0.01771	-0.01516	-0.01959	-0.00152
TabPFN	Wine Quality	-0.02148	-0.01159	-0.01104	-0.01867	-0.00763

Table 26: Reserve leave-one-dataset-out P3−P2 contrasts. The omitted set is one of four independent reserve datasets; fallback rates are percentages.

Model	Omitted dataset	Fallback rate	ΔHarm	$\Delta(h > 0.05)$	ΔP95	ΔCVaR90	ΔGain
TabICL	Breast Cancer	4.38%	-0.01224	-0.00703	-0.01647	-0.01700	+0.00082
TabICL	Phishing	5.00%	-0.01797	-0.00625	-0.01320	-0.01541	+0.00059
TabICL	Heart Disease	3.59%	-0.01250	-0.00286	-0.00701	-0.01001	+0.00012
TabICL	Raisin	4.22%	-0.01979	-0.00729	-0.01720	-0.01621	+0.00069
TabPFN	Breast Cancer	4.38%	-0.01224	-0.00599	-0.01032	-0.01405	-0.00035
TabPFN	Phishing	5.00%	-0.01250	-0.00573	-0.01092	-0.01217	-0.00057
TabPFN	Heart Disease	3.59%	-0.01120	-0.00339	-0.00352	-0.00380	-0.00073
TabPFN	Raisin	4.22%	-0.01172	-0.00521	-0.01238	-0.01501	+0.00015

Table 27: Pre-specified extreme-query checks. The exploratory panel deletes one maximum-absolute-gain query within each unit; the reserve panel reports the P3−P2 statistics available after the same prespecified deletion.

Evidence set	Model	Policy/quantity	Full value	After deletion	Change
Exploratory	TabICL	P1 gain	0.040521	0.056568	+0.016048
Exploratory	TabICL	P2 gain	0.023213	0.030487	+0.007275
Exploratory	TabICL	P3 gain	0.020075	0.027331	+0.007256
Exploratory	TabPFN	P1 gain	0.009302	0.002941	-0.006361
Exploratory	TabPFN	P2 gain	0.008925	-0.008882	-0.017807
Exploratory	TabPFN	P3 gain	0.004018	-0.009214	-0.013233

Reserve mini-panel: after-deletion P3−P2 endpoints

Backbone	ΔHarm	$\Delta(h > 0.05)$	ΔCVaR90
TabICL	-0.01392	-0.00412	-0.00390
TabPFN	-0.01020	-0.00353	-0.00441

For the reserve sensitivity analysis, only the displayed post-deletion contrasts are reported; corresponding full-sample contrasts and deletion-induced changes are not reported in this panel.

D.10 Conditional Random Fallback and Opportunity Thresholds

Random fallback uses seed 20260809 and keeps the observed fallback count fixed within each cell. The hindsight alternative, ORACLE_HARM_FALLBACK, applies fallback to the queries with the largest realized harms. It provides a non-deployable upper bound because it requires the query outcomes.

Table 28: Comparison with random fallback at the same fallback rate, using saved predictions. Intervals are the 2.5%–97.5% Monte Carlo quantiles over 10,000 conditional assignments per evidence-set/backbone combination. The P3 percentile gives its position in the conditional random-assignment distribution. A lower percentile is favorable for harm measures, while a higher percentile is favorable for gain.

Set	Backbone	Endpoint	P3	Random mean	2.5%–97.5%	P3−random	P3 percentile
Exploratory	TabICL	Harm	0.4082	0.4037	[0.4017, 0.4058]	+0.0045	100.00%
Exploratory	TabICL	$h > 0.05$	0.3495	0.3470	[0.3451, 0.3489]	+0.0025	99.42%
Exploratory	TabICL	P95	0.9109	0.9066	[0.8968, 0.9153]	+0.0043	81.64%
Exploratory	TabICL	CVaR90	1.1015	1.1002	[1.0897, 1.1095]	+0.0013	58.71%
Exploratory	TabICL	Gain	0.0201	0.0224	[0.0199, 0.0249]	−0.0023	3.62%
Exploratory	TabPFN	Harm	0.4965	0.4922	[0.4900, 0.4942]	+0.0043	100.00%

Table 28 – continued from previous page

Set	Backbone	Endpoint	P3	Random mean	2.5%–97.5%	P3−random	P3 percentile
Exploratory	TabPFN	$h > 0.05$	0.3945	0.3885	[0.3864, 0.3905]	+0.0061	100.00%
Exploratory	TabPFN	P95	0.5620	0.5647	[0.5585, 0.5696]	−0.0027	17.22%
Exploratory	TabPFN	CVaR90	0.6794	0.6857	[0.6788, 0.6913]	−0.0063	3.58%
Exploratory	TabPFN	Gain	0.0040	0.0091	[0.0045, 0.0120]	−0.0051	1.66%
Reserve	TabICL	Harm	0.4193	0.4168	[0.4143, 0.4193]	+0.0025	97.60%
Reserve	TabICL	$h > 0.05$	0.2068	0.2035	[0.2012, 0.2057]	+0.0034	99.93%
Reserve	TabICL	P95	0.4037	0.4018	[0.3915, 0.4110]	+0.0019	63.26%
Reserve	TabICL	CVaR90	0.5177	0.5185	[0.5093, 0.5258]	−0.0008	39.74%
Reserve	TabICL	Gain	0.0028	0.0021	[0.0005, 0.0038]	+0.0007	79.99%
Reserve	TabPFN	Harm	0.3164	0.3145	[0.3119, 0.3170]	+0.0019	94.38%
Reserve	TabPFN	$h > 0.05$	0.1264	0.1255	[0.1236, 0.1273]	+0.0009	83.97%
Reserve	TabPFN	P95	0.1659	0.1672	[0.1620, 0.1718]	−0.0013	28.21%
Reserve	TabPFN	CVaR90	0.2567	0.2592	[0.2519, 0.2644]	−0.0024	21.23%
Reserve	TabPFN	Gain	0.0100	0.0098	[0.0089, 0.0108]	+0.0002	63.46%

Positive P3−random values are worse for harm endpoints and better for paired gain. These conditional comparisons describe random fallback within the observed queries, not performance on new datasets.

Table 29: Overall Bank-16 opportunity prevalence at the evaluated post-hoc thresholds. The denominator is the number of query–backbone observations in each evidence stratum.

ε	Exploratory			Reserve		
	Count	Denom.	Prevalence	Count	Denom.	Prevalence
0	19,268	20,480	94.08%	9,922	10,240	96.89%
10^{-4}	19,267	20,480	94.08%	9,896	10,240	96.64%
10^{-3}	19,251	20,480	94.00%	9,645	10,240	94.19%
0.01	18,938	20,480	92.47%	8,243	10,240	80.50%
0.05	16,934	20,480	82.69%	6,076	10,240	59.34%

Conditional on $\delta^{\text{oracle}} > 0$, exploratory values are Q25 = 0.0971, median = 0.2633, Q75 = 0.6287, Q90 = 1.0630, and Q95 = 1.4046. Reserve values are Q25 = 0.0202, median = 0.0856, Q75 = 0.2363, Q90 = 0.5330, and Q95 = 0.7971. These quantiles describe the loss scale; they do not identify thresholds of practical importance.

Table 30: Dataset-by-backbone opportunity prevalence at the evaluated post-hoc thresholds.

Evidence	Backbone	Dataset	$\varepsilon = 0$	10^{-4}	10^{-3}	0.01	0.05
Exploratory	TabICL	Bank Marketing	97.93%	97.93%	97.85%	95.66%	74.02%
Exploratory	TabPFN	Bank Marketing	92.34%	92.30%	92.03%	87.03%	58.63%
Exploratory	TabICL	Covertype	91.88%	91.88%	91.80%	91.17%	88.01%
Exploratory	TabPFN	Covertype	91.09%	91.09%	91.05%	89.88%	85.23%
Exploratory	TabICL	Online Shoppers	96.29%	96.29%	96.25%	95.31%	88.59%
Exploratory	TabPFN	Online Shoppers	95.98%	95.98%	95.90%	95.12%	88.40%
Exploratory	TabICL	Wine Quality	93.20%	93.20%	93.20%	92.42%	89.53%
Exploratory	TabPFN	Wine Quality	93.95%	93.95%	93.91%	93.16%	89.06%
Reserve	TabICL	Breast Cancer	94.45%	92.89%	80.47%	43.98%	21.80%
Reserve	TabPFN	Breast Cancer	96.80%	96.48%	93.59%	71.41%	37.19%
Reserve	TabICL	Phishing	96.56%	96.48%	93.75%	75.70%	50.55%
Reserve	TabPFN	Phishing	98.83%	98.83%	98.05%	82.11%	55.08%
Reserve	TabICL	Heart Disease	91.64%	91.64%	91.25%	89.22%	71.56%
Reserve	TabPFN	Heart Disease	98.91%	98.83%	98.83%	97.73%	84.77%
Reserve	TabICL	Raisin	98.91%	98.91%	98.75%	93.20%	75.55%
Reserve	TabPFN	Raisin	99.06%	99.06%	98.83%	90.63%	78.20%

D.11 Fresh-Holdout Single-Call Evaluation

P2 and P3 exceed random conversion on nine of ten fresh-holdout datasets per backbone, except TabICL P3, which does so on eight; P1's results are mixed (Table 31). These conversion gains measure whole-policy success, not recovery of the hindsight-best candidate or full ranking. The controls below examine rule-identity assignment relative to marginal preference. Design and estimands are in Appendices A.1 and B.5.

The P2/P3 utility aggregation and bootstrap analysis were specified after inference began, before local outcome evaluation. This supplementary analysis supports positive average single-call utility in the evaluated setting. For each policy, gains are positive on six datasets for TabPFN and nine for TabICL. Table 32 reports the dataset effects.

Table 31: Fresh-holdout single-call results. The upper panel reports the difference from uniform random selection in conditional conversion, in percentage points. The lower panel reports supplementary NLL gains over the fixed reference, using all queries. Both panels give datasets equal weight. The pooled ratio-of-sums is a separate descriptive summary, unlike the primary weighting used here. Intervals are pointwise 95% dataset-bootstrap intervals, not simultaneous intervals. W/T/L counts datasets with positive, zero, and negative effects.

Backbone / policy	Mean	95% interval	W/T/L
Conditional conversion minus random (pp)			
TabPFN / P1	+1.89	[−7.35, +11.36]	5/0/5
TabPFN / P2	+8.89	[+5.64, +11.69]	9/0/1
TabPFN / P3	+6.53	[+3.57, +9.16]	9/0/1
TabICL / P1	+7.55	[−0.11, +14.41]	8/0/2
TabICL / P2	+8.75	[+5.17, +12.42]	9/0/1
TabICL / P3	+5.69	[+2.60, +9.11]	8/0/2
Supplementary utility (NLL)			
TabPFN / P1	−0.004286	[−0.015061, +0.002738]	3/0/7
TabPFN / P2	+0.043222	[+0.000423, +0.099405]	6/0/4
TabPFN / P3	+0.042921	[+0.000277, +0.098807]	6/0/4
TabICL / P1	+0.026193	[−0.045051, +0.094016]	8/0/2
TabICL / P2	+0.067826	[+0.026547, +0.115424]	9/0/1
TabICL / P3	+0.061658	[+0.024181, +0.104427]	9/0/1

Table 32: Fresh-holdout dataset effects over two seeds. Conversion is policy minus uniform random (pp); utility is supplementary all-query NLL gain over the fixed reference. Signs are descriptive, not dataset-level significance tests.

Dataset	Backbone	Conversion minus random (pp)			Utility (NLL)	
		P1	P2	P3	P2	P3
kc2	TabPFN	−17.62	−1.97	−3.34	+0.009208	+0.008107
kc2	TabICL	−18.46	−0.11	−1.12	+0.000942	+0.001069
pc4	TabPFN	−0.54	+6.25	+4.40	−0.005177	−0.004741
pc4	TabICL	−2.45	+6.70	+3.91	+0.021855	+0.018713
pc3	TabPFN	+1.79	+15.66	+11.04	+0.225843	+0.225424
pc3	TabICL	+11.14	+1.38	−0.24	−0.004954	−0.000537
analcatdata_dmft	TabPFN	−12.15	+5.64	+4.82	+0.021724	+0.020373
analcatdata_dmft	TabICL	+1.01	+3.19	+1.88	+0.035565	+0.032890
vehicle	TabPFN	+12.01	+14.39	+13.10	+0.020023	+0.021903
vehicle	TabICL	+13.50	+19.14	+16.12	+0.118653	+0.111860
ilpd	TabPFN	+6.34	+10.03	+3.27	−0.011673	−0.014560
ilpd	TabICL	+2.59	+11.64	+3.02	+0.011283	+0.002387
pendigits	TabPFN	+23.91	+10.79	+9.88	+0.178454	+0.176943
pendigits	TabICL	+21.84	+13.44	+11.84	+0.185640	+0.175816
cylinder-bands	TabPFN	−17.91	+12.04	+9.62	−0.000247	−0.000570
cylinder-bands	TabICL	+8.37	+8.16	+4.81	+0.050668	+0.047368
churn	TabPFN	−4.05	+6.60	+4.86	+0.000058	+0.000107
churn	TabICL	+16.72	+14.46	+10.44	+0.052529	+0.050671
optdigits	TabPFN	+27.17	+9.45	+7.70	−0.005997	−0.003772
optdigits	TabICL	+21.29	+9.47	+6.26	+0.206075	+0.176345

Post-hoc targeted controls. We next ask whether the beyond-random result includes value from assigning candidate identities to particular queries, and whether TabPFN utility depends on the original fixed reference. These two later analyses use only saved predictions, labels, and policy selections. They do not replace the primary results in Tables 31–32.

Frequency-matched identity control. Within each dataset–seed–backbone–policy cell, let $f(c) = 256^{-1} \sum_q \mathbf{1}[\pi(q) = c]$ be the frequency of candidate identity c among all 256 saved policy selections. Define

$$L_{\text{freq}}(q) = \sum_c f(c) L(q, c), \qquad g_{\text{assign}}(q) = L_{\text{freq}}(q) - L(q, \pi(q)).$$

The frequency uses neither labels nor losses and is not re-estimated on the bank-positive subset. It preserves each policy's marginal identity preference while removing query-to-identity correspondence in expectation. The loss $L(q, c)$ always comes from the current query's saved output for rule c; no other query's physical context is transferred. The control therefore retains query dependence internal to a candidate rule. It is a diagnostic comparator, not a causal identification procedure.

For the conversion analogue, retain the original reference $L_{\text{ref}}(q)$ and $O(q) = \mathbf{1}[\min_c L(q, c) < L_{\text{ref}}(q)]$. The control's probability of beating that reference is

$$p_{\text{freq}}(q) = \sum_c f(c) \mathbf{1}[L(q, c) < L_{\text{ref}}(q)].$$

Within each dataset, sum policy successes and control expected successes over bank-positive queries across both seeds, divide each by their combined bank-positive count, and take the policy-minus-control difference. NLL gains instead average all queries within seed, then average the two seeds. Both endpoints give the ten datasets equal weight. Intervals below are pointwise 95% paired dataset-bootstrap intervals from 10,000 repetitions, using the original bootstrap seed and percentile definition. W/T/L counts positive, zero, and negative dataset effects. The reassignment expectation is computed exactly; no permutation or sign test is performed.

Backbone	Policy	Mean	95% interval	W/T/L
Assignment gain (NLL)				
TabPFN	P2	$+0.000495$	$[-0.000121, +0.001352]$	3/6/1
TabPFN	P3	$+0.002635$	$[-0.000455, +0.006807]$	7/0/3
TabICL	P2	$+0.000044$	$[-0.000251, +0.000374]$	2/6/2
TabICL	P3	-0.003912	$[-0.008325, -0.000473]$	2/0/8
Conversion minus frequency-matched control (pp)				
TabPFN	P2	$+0.0762$	$[-0.0662, +0.2691]$	2/6/2
TabPFN	P3	-0.1946	$[-0.8459, +0.3781]$	5/0/5
TabICL	P2	-0.0456	$[-0.1566, +0.0173]$	2/6/2
TabICL	P3	-0.5722	$[-1.4773, +0.2039]$	3/0/7

TabICL P3 has higher mean loss than the frequency-matched expectation, with its NLL interval below zero. The remaining NLL intervals and all conversion intervals cross zero. Positive added assignment value remains unestablished; crossing intervals leave the direction unresolved rather than establish equivalence. P2 selections agree across backbones and choose CLASS_BALANCED_KNN on 5,080/5,120 queries (99.21875%) per backbone. On kc2, pc4, pc3, vehicle, ilpd, and optdigits, P2 selects that identity on every query of both seeds, making its frequency-matched contrasts exactly zero on those datasets. This is consistent with strong marginal candidate-rule preference behind the beyond-random result, without attributing all gains to it. CLASS_BALANCED_KNN can itself construct different contexts for different queries; the result does not imply that query information or rule-internal conditioning has no value.

TabPFN K64 alternate-reference sensitivity. The specified alternative is RANDOM_SLOT_6, the K64-only development sensitivity winner in Appendix D.8, not a reference selected using fresh outcomes. We evaluate the unchanged P2/P3 selections against it. In particular, P3 fallback still returns the original STANDARDIZED_KNN reference; only the evaluation comparator changes. Mean utility is $L_{\mathrm{alt}}(q) - L(q, \pi(q))$, averaged over all queries, the two seeds, and then datasets as above.

For conversion, redefine $O_{\mathrm{alt}}(q) = \mathbf{1}[\min_c L(q,c) < L_{\mathrm{alt}}(q)]$ and the uniform-random success probability $16^{-1}\sum_c \mathbf{1}[L(q,c) < L_{\mathrm{alt}}(q)]$. Combine the two seeds' numerators and denominators within dataset before computing each policy-minus-random contrast. This sensitivity uses uniform random choice, not the frequency-matched comparator. The same dataset-bootstrap procedure gives:

Policy	Mean	95% interval	W/T/L
Utility versus RANDOM_SLOT_6 (NLL)			
P2	$+0.061906$	$[+0.020844, +0.105318]$	8/0/2
P3	$+0.061606$	$[+0.020168, +0.105487]$	8/0/2
Conversion minus uniform random choice (pp)			
P2	$+9.8246$	$[+5.8983, +13.4495]$	9/0/1
P3	$+9.7694$	$[+5.6980, +13.5694]$	9/0/1

The alternative defines 4,795/5,120 queries as bank-positive (93.652344%). Both utility and conversion intervals remain positive for both policies under this specified reference. Changing the comparator shifts both policies' mean utility by the same amount; it does not improve their selections or redefine the primary reference. The altered bank-positive set also prevents interpreting a larger conversion contrast as improved assignment.

P2 concentration and fixed-rule control. A further post-hoc analysis distinguishes a strict coverage-score preference from deterministic selection among tied minima. We used the saved candidate scores and the implemented lexicographic identity tie-break, reproducing all 5,120 query decisions; scores and selections agreed across backbones. Ties use exact floating-point equality, without a tolerance. Counts below refer to dataset–seed–query decisions, counted once across backbones.

Minimum-score outcome	Count	Percentage
CLASS_BALANCED_KNN uniquely best	9	0.175781%
CLASS_BALANCED_KNN selected from tied minima	5,071	99.042969%
Another identity uniquely best	1	0.019531%
Another identity selected from tied minima	39	0.761719%

Exact ties occur on 5,110/5,120 queries (99.804688%); tied sets have median size eight and maximum size sixteen. Of P2's 5,080 CLASS_BALANCED_KNN selections, 5,071 (99.82%) result from tied minima resolved lexicographically, rather than a unique score minimum. Thus the concentration mainly reflects the implemented tie-break among minimum-score candidates. This explains the recorded identity choices, not the causal origin of the tied scores or their outcome performance.

We also compare P2 directly with always using the existing Bank-16 CLASS_BALANCED_KNN rule. Positive NLL gain is $L_{\mathrm{CBKNN}} - L_{\mathrm{P2}}$; positive conversion gain is $100(C_{\mathrm{P2}} - C_{\mathrm{CBKNN}})$. Conversion retains the original backbone-specific reference and bank-positive set. NLL averages queries within seed, the two seeds within dataset, then the ten datasets equally; conversion first combines the two seeds' success counts and denominators. The following pointwise 95% intervals use 10,000 paired dataset-bootstrap repetitions with the original seed and percentile definition. W/T/L counts dataset effects.

Backbone	Mean gain	95% interval	W/T/L
NLL gain over fixed CLASS_BALANCED_KNN			
TabPFN	$+0.000112$	$[-0.000519, +0.000950]$	2/6/2
TabICL	-0.000317	$[-0.000867, +0.000141]$	1/6/3
Conditional conversion gain over the fixed rule (pp)			
TabPFN	$+0.0222$	$[-0.1001, +0.1441]$	2/7/1
TabICL	-0.1073	$[-0.2620, 0.0000]$	0/8/2

The direct comparison does not establish an incremental P2 advantage over the fixed rule on either endpoint or backbone. Both NLL intervals and the TabPFN conversion interval cross zero; the TabICL conversion interval has an upper endpoint exactly at zero. These are not equivalence tests. P2 switches away from CLASS_BALANCED_KNN on only 40/5,120 queries: 11 in analcatdata_dmft, one in pendigits, 27 in cylinder-bands, and one in churn. The other 5,080 losses are exactly identical, so these switches account for the entire observed NLL difference.

P2 and CLASS_BALANCED_KNN favor classwise proximity but do not optimize an identical metric. P2 evaluates candidate contexts using source-fit numeric scaling and categorical mismatch indicators. CLASS_BALANCED_KNN is constructed using support-fitted scaling and Euclidean distances on the resulting features. Its physical context remains query-dependent even when its rule identity is fixed. The comparison concerns the added increment from routing among candidate-rule identities, which remains unestablished.

Score bottlenecks and tie-breaking sensitivity. These post-hoc diagnostics reuse the same ten fresh-holdout datasets, two seeds, Bank-16/K64/MCAR15 contexts, and saved predictions. Replaying the original geometry, including source-fit numeric scaling, categorical mismatch distances, and the original numeric precision, exactly reproduces every saved P1/P2 score and all 5,120 selections for each policy. Context rows, their order, and scores agree across backbones; geometry counts therefore count each dataset–seed–query once. All score ties use exact equality, with no near-tie tolerance. Query outcomes enter only the subsequent loss comparisons.

For query q, candidate support set S_c, and support class k, define

$$d_k(q,c) = \min_{i \in S_c : y_i = k} d(q, x_i), \qquad s(q,c) = \max_k d_k(q,c), \qquad M(q) = \arg\min_c s(q,c).$$

An absent class gives infinite distance; all observed minimum scores are finite. The bottleneck class set is $B_c(q) = \{k : d_k(q,c) = s(q,c)\}$. For each class retain every nearest support row, $R_{c,k}(q) = \{i \in S_c : y_i = k,\ d(q,x_i) = d_k(q,c)\}$, and define the bottleneck row set $R_c(q) = \bigcup_{k \in B_c(q)} R_{c,k}(q)$. Thus neither class nor row ties are resolved arbitrarily for this diagnostic.

Among candidates in $M(q)$, identical sets, a nonempty intersection across *all* sets, and overlap of at least one pair are distinct properties. In particular, an empty global intersection does not imply pairwise disjointness. The following percentages use the 5,110 P2 tied queries as denominator.

Bottleneck-set property	Classes	Support rows
All sets identical	91.4481%	89.6282%
Nonempty intersection across all minima	100.0000%	93.7182%
Nonidentical sets with at least one overlapping pair	8.5519%	10.3718%
Every pair disjoint	0.0000%	0.0000%
Empty intersection across all minima	0.0000%	6.2818%

All tied queries share a bottleneck class, and most also share a bottleneck row, even though their contexts differ. The row-level pattern is heterogeneous: only 37.5% of tied queries in analcatdata_dmft have a row common to all minima, versus 99.8047% in ilpd and 100% in the other datasets. The score retains only the largest classwise nearest distance, leaving other context differences unresolved. This explains equality of the scores, not which specific support rows cause differences in predictions.

For each unordered pair $c, c' \in M(q)$, support-set Jaccard similarity is $|S_c \cap S_{c'}|/|S_c \cup S_{c'}|$. Across 168,393 pairs, its pair-weighted median is 0.347368 and 87.9158% of pairs have similarity below 0.5. Each pair has equal weight, so queries with larger $M(q)$ contribute more pairs; these are not query-weighted summaries. Jaccard ignores support order: a value of one establishes equal row sets, not equal ordered contexts or predictions.

P1 first minimizes nearest distance, then mean distance, and finally lexicographic candidate identity. P2 has no mean-distance secondary criterion. The full pre-identity P1 score is the ordered tuple of its two distances; identity strings are excluded from distinct-score counts. The following summaries use all 5,120 queries.

Decision criterion	P1	P2
Primary exact-minimum tie rate	100%	99.804688%
Median primary minimum-set size	10	8
Median distinct primary scores	4	5
Ties after all pre-identity scores	176/5,120	5,110/5,120
Final pre-identity tie rate	3.4375%	99.804688%
Median distinct full score tuples	16	5

Thus primary ties alone do not characterize a policy's unresolved choices. P1's secondary criterion removes most ties on the same candidate bank; this describes discriminability, not superior predictive ranking or a causal effect of score design. P1's 176 remaining tied queries and P2's 5,110 tied queries are different subsets, so their predictive-dispersion summaries are not a paired effect test.

Dataset-level geometry. n_{tie} counts P2 tied queries; the P2 tie rate and median $|M|$ use all 512 queries per dataset. Shared-row percentages condition on n_{tie}; the final P1 tie rate uses all 512 queries. P1's primary tie rate is 100% and its median distinct full score count is 16 in every dataset.

Dataset	n_{tie}	P2 ties (%)	Median $\lvert M\rvert$	Shared row (%)	P1 ties (%)
kc2	512	100.00	9	100.00	6.64
pc4	512	100.00	9	100.00	0.78
pc3	512	100.00	9	100.00	1.37
analcatdata_dmft	512	100.00	14	37.50	4.30
vehicle	512	100.00	8	100.00	8.20
ilpd	512	100.00	8	99.80	0.00
pendigits	505	98.63	5	100.00	6.84
cylinder-bands	511	99.80	7	100.00	6.05
churn	512	100.00	9	100.00	0.00
optdigits	510	99.61	6	100.00	0.20

Predictions within tied sets. For each query, compute Jensen–Shannon divergence over unordered pairs in $M(q)$ and average those pair values before summarizing across queries. We reuse the predictive-centrality JS definition: clip probabilities to $[10^{-12}, 1]$, use natural logarithms and their equal-weight midpoint, with no additional normalization. Class dimensions follow the saved model probabilities. NLL uses the saved candidate losses. For each tied query, its NLL range is $\max_{c\in M(q)} L(q,c) - \min_{c\in M(q)} L(q,c)$. The following query-weighted summaries exclude singleton minima and each use 5,110 queries per backbone.

Backbone	Query statistic	Mean	Median	25th pct.	75th pct.
TabPFN	Mean pairwise JS	0.0196771	0.00785263	0.0035782	0.0216980
TabPFN	NLL range	0.5066230	0.32816675	0.1359265	0.6798940
TabICL	Mean pairwise JS	0.0147260	0.00714242	0.0034746	0.0186321
TabICL	NLL range	0.4615654	0.30505004	0.1284424	0.6486167

Every P2 tied set contains differing predictions and losses on both backbones; this does not mean that every candidate pair differs. NLL range is a hindsight spread, not recoverable utility or a tie-break effect. Dataset-level medians below retain the distinction between pair-weighted Jaccard and query-weighted model dispersion.

Dataset	Jaccard	PFN JS	PFN range	ICL JS	ICL range
kc2	0.347368	0.003391	0.144866	0.003687	0.136326
pc4	0.347368	0.003209	0.089990	0.003907	0.113284
pc3	0.347368	0.003827	0.121802	0.002753	0.114528
analcatdata_dmft	0.333333	0.007037	0.503455	0.011183	0.623630
vehicle	0.347368	0.022036	0.526405	0.016458	0.510947
ilpd	0.347368	0.004323	0.268932	0.004386	0.256409
pendigits	0.347368	0.042248	0.541455	0.023752	0.494989
cylinder-bands	0.347368	0.010730	0.507373	0.011157	0.475113
churn	0.347368	0.006517	0.181415	0.004212	0.158638
optdigits	0.347368	0.059765	0.694753	0.037515	0.577070

Random choice among minima. Keep the formal candidate slots and scores unchanged, without deduplication or reweighting. Let $\pi_{\text{lex}}(q)$ be the original lexicographic choice and let r denote the original backbone-specific reference. The uniform-minimum comparator has exact expected loss and reference-beating probability

$$L_{\text{rand}}(q) = \frac{1}{|M(q)|}\sum_{c\in M(q)} L(q,c), \qquad a_{\text{rand}}(q) = \frac{1}{|M(q)|}\sum_{c\in M(q)} \mathbf{1}[L(q,c) < L_r(q)].$$

This is the expectation from selecting one candidate, not NLL after probability averaging. Singleton minima give identical actions. Positive NLL gain is $L_{\text{rand}} - L(q, \pi_{\text{lex}}(q))$, averaged over queries, then seeds, then equally over datasets. With the original bank-positive indicator $O(q) = \mathbf{1}[\min_c L(q,c) < L_r(q)]$, conversion within each dataset is

$$C_{\text{rand}} = \frac{\sum_{s,q} O(q) a_{\text{rand}}(q)}{\sum_{s,q} O(q)}, \qquad C_{\text{lex}} = \frac{\sum_{s,q} O(q)\mathbf{1}[L(q,\pi_{\text{lex}}(q)) < L_r(q)]}{\sum_{s,q} O(q)}.$$

Here the sums combine both seeds within dataset; reported conversion gain is the equal-dataset mean of $100(C_{\text{lex}} - C_{\text{rand}})$ in percentage points. References remain STANDARDIZED_KNN for TabPFN and RANDOM_SLOT_6 for TabICL. All 5,120 queries enter the NLL comparison, including singleton minima.

The following saved intervals use 10,000 paired dataset-bootstrap repetitions over the ten datasets, seed 20260918, and linear percentile endpoints. They are pointwise 95% intervals, not a simultaneous guarantee. Neither queries, seeds, nor backbones are treated as independent replicates; W/T/L counts dataset effects.

Backbone	Endpoint	Gain	Pointwise 95% interval	W/T/L
TabPFN	NLL	$+0.036484$	$[+0.013604, +0.063817]$	10/0/0
TabPFN	Conversion (pp)	$+12.145697$	$[+9.763991, +15.072717]$	10/0/0
TabICL	NLL	$+0.031475$	$[+0.009001, +0.058551]$	9/0/1
TabICL	Conversion (pp)	$+10.715501$	$[+7.921752, +13.158349]$	10/0/0

The original lexicographic rule outperforms uniform choice among P2 minima on both endpoints and backbones in this evaluation. The next block shows the corresponding dataset effects; gains retain the same direction as above.

Dataset	PFN NLL	ICL NLL	PFN conv. (pp)	ICL conv. (pp)
kc2	+0.011265	+0.012427	+9.777805	+12.389394
pc4	+0.002055	+0.010003	+11.739190	+10.043707
pc3	+0.008832	+0.007924	+23.036500	+14.572863
analcatdata_dmft	+0.002305	-0.005799	+7.437746	+1.781113
vehicle	+0.038154	+0.063637	+13.476397	+12.765961
ilpd	+0.002017	+0.000815	+14.369818	+11.507127
pendigits	+0.121412	+0.122856	+10.627967	+16.043903
cylinder-bands	+0.044172	+0.015247	+14.218264	+7.129989
churn	+0.032455	+0.010661	+8.623349	+14.416233
optdigits	+0.102172	+0.076977	+8.149935	+6.504719

The implemented tie-break changes measured P2 performance. Together with the frequency-matched and fixed-rule controls, it separates whole-policy performance, score discriminability, tie-breaking sensitivity, and identity-assignment value. This result concerns these geometric policies and the same ten-dataset Bank-16/K64/MCAR15 evaluation; it neither attributes all gains to tie-breaking nor validates lexicographic selection in other banks or policies.

D.12 Post-hoc P2 score-smoothing sensitivity

This post-hoc sensitivity uses the same ten fresh-holdout datasets, two seeds, Bank-16/K64/MCAR15 contexts, and saved predictions and losses; no new TFM inference was run. Original P2 scores candidates by $s_{\max}(q, c) = \max_k d_k(q, c)$, with d_k as defined above. For each query, let a_q be the median of strictly positive finite classwise distances across candidates and source classes, or 1 if none exists, and set $z_k = d_k/a_q$. The score and $\beta = 10$ were fixed, and selections saved, before downstream outcomes were read. This is not independent confirmation. The sensitivity score is

$$s_{\text{smooth}}(q, c) = \frac{a_q}{\beta} \left[\text{logsumexp}_k(\beta z_k(q, c)) - \log C\right], \qquad \beta = 10,$$

where C is the number of source classes. Log-sum-exp is evaluated stably; absent classes give $+\infty$, and remaining exact ties use the original lexicographic identity rule.

Table 33: Post-hoc P2 smooth-max sensitivity on the fresh holdout. Equal-dataset estimates and pointwise 95% paired dataset-bootstrap intervals; positive NLL increments favor the named policy.

Quantity	Original P2	SmoothMax ($\beta = 10$)
Exact score ties	5,110/5,120 (99.80%)	4,532/5,120 (88.52%)
CBKNN selected	5,080/5,120 (99.22%)	5,050/5,120 (98.63%)
Agreement with original P2	—	5,082/5,120 (99.26%)
Median minimum-set size	8	6
TabPFN conversion vs. random (pp)	$+8.889$ $[+5.642, +11.686]$	$+8.804$ $[+5.635, +11.544]$
TabICL conversion vs. random (pp)	$+8.747$ $[+5.169, +12.424]$	$+8.728$ $[+5.141, +12.421]$
TabPFN frequency-matched NLL	$+0.000495$ $[-0.000121, +0.001352]$	-0.000238 $[-0.001114, +0.000821]$
TabICL frequency-matched NLL	$+0.000044$ $[-0.000251, +0.000374]$	-0.000202 $[-0.000950, +0.000336]$
TabPFN NLL vs. fixed CBKNN	$+0.000112$ $[-0.000519, +0.000950]$	-0.000786 $[-0.002274, +0.000535]$
TabICL NLL vs. fixed CBKNN	-0.000317 $[-0.000867, +0.000141]$	-0.000562 $[-0.001514, +0.000166]$

SmoothMax reduces exact ties but leaves them common; 99.26% of original selections remain unchanged and beyond-random conversion is nearly identical. Neither the frequency-matched nor fixed-rule comparison establishes a positive added rule-identity assignment increment. In a separate paired comparison, SmoothMax's NLL gain over original P2 is -0.000898 $[-0.001870, -0.000092]$ for TabPFN and -0.000245 $[-0.000651, +0.000045]$ for TabICL: the first pointwise interval supports a small loss increase, while the second crosses zero. The sensitivity concerns this scoring rule, candidate bank, and holdout datasets.

E Predictive-Centrality Hypothesis, Replication, Formal Test, and Fresh-Holdout Evaluation

E.1 Exploratory model-aware pilot

In leave-one-dataset-out evaluation, the conservative learned router has mean NLL gain -0.016067 against the strongest pilot comparator and wins in 37.5% of dataset–model comparisons. This pilot led us to consider predictive centrality, but it did not validate the rule. We subsequently fixed PC to select the prediction with the smallest mean pairwise Jensen–Shannon divergence, breaking ties by the lowest slot. The rule does not use query outcomes.

E.2 Disjoint replication

On 18 disjoint datasets, gain over SINGLETON_MMD is +0.012502 for TabPFN and +0.005162 for TabICL. The overall mean is +0.008832. Backbone-specific cell W/T/L counts are 12/0/6 and 11/0/7, respectively. The LODO range [+0.003841, +0.011459] describes the effect of omitting each dataset, rather than statistical uncertainty around the mean. The secondary comparison with averaging gives a mean gain of -0.000238; that point estimate alone does not establish that PC is worse.

E.3 Original formal test

The original formal test uses MCAR15 and K = 64, with 16 candidates for the primary comparison and 4, 8, and 32 for prespecified bank-size checks. We specified the methods and comparisons before examining test outcomes and finalized predictions and selections before accessing query labels. CRUMB was selected on development data as the one-context comparator. It uses an unlabeled query batch, unlike the single-query R0 policies, and is different from the replication comparator.

At B16, the original results are:

Reporting level	Comparator	Mean gain	Original 95% CI	W/T/L
TabPFN	CRUMB	+0.009311	[+0.002447, +0.015497]	10/0/2
TabPFN	Averaging	+0.000675	[−0.000435, +0.001788]	9/0/3
TabICL	CRUMB	+0.012325	[+0.006802, +0.019409]	11/0/1
TabICL	Averaging	+0.000054	[−0.000754, +0.000767]	8/0/4
Pooled secondary	CRUMB	+0.010818	[+0.005774, +0.016186]	21/0/3
Pooled secondary	Averaging	+0.000365	[−0.000349, +0.001119]	17/0/7

For each backbone, W/T/L counts the 12 dataset-level effects. The original pooled counts combine 24 dataset–backbone comparisons, but the bootstrap still resamples 12 datasets. The complete original-bank results retain the B4 interval crossing zero against CRUMB. Post-hoc t_{11} intervals and LODO mean ranges remain separate sensitivity summaries and do not replace the original bootstrap intervals.

Table 34: Bank-16 formal-test effects by dataset and backbone. Gain is comparator NLL minus PC NLL; positive values favor PC. The 24 dataset–model comparisons give mean gains of +0.010818 over CRUMB and +0.000365 over averaging, with W/T/L counts of 21/0/3 and 17/0/7. Inference resamples the 12 datasets. The lower panel gives later small-cluster sensitivity checks, separately from the primary bootstrap intervals.

Dataset	Backbone	PC NLL	CRUMB NLL	Gain vs CRUMB	Avg. NLL	Gain vs avg.
Bank Customer Churn	TabICL	0.490442	0.493729	+0.003287	0.490055	-0.000387
Bank Customer Churn	TabPFN	0.487750	0.485373	-0.002378	0.485420	-0.002330
Blood transfusion	TabICL	0.523497	0.531536	+0.008039	0.523757	+0.000260
Blood transfusion	TabPFN	0.517699	0.523779	+0.006080	0.517951	+0.000252
COIL2000	TabICL	0.245159	0.255825	+0.010666	0.245641	+0.000482
COIL2000	TabPFN	0.242201	0.263560	+0.021359	0.243047	+0.000846
Diabetes	TabICL	0.527480	0.543006	+0.015526	0.528100	+0.000620
Diabetes	TabPFN	0.513040	0.525125	+0.012085	0.513841	+0.000801
Diabetes130US	TabICL	0.291239	0.297063	+0.005824	0.292332	+0.001094
Diabetes130US	TabPFN	0.279755	0.280318	+0.000563	0.280104	+0.000349
Is-this-a-good-customer	TabICL	0.324524	0.331190	+0.006666	0.325813	+0.001288
Is-this-a-good-customer	TabPFN	0.317115	0.321500	+0.004385	0.319891	+0.002776
Marketing Campaign	TabICL	0.359971	0.367728	+0.007757	0.360346	+0.000375
Marketing Campaign	TabPFN	0.371756	0.382228	+0.010472	0.369902	-0.001854
Maternal health risk	TabICL	0.748398	0.759096	+0.010698	0.745685	-0.002713
Maternal health risk	TabPFN	0.764264	0.744894	-0.019370	0.766252	+0.001989
NATICUSdroid	TabICL	0.295796	0.339684	+0.043888	0.293477	-0.002319
NATICUSdroid	TabPFN	0.308620	0.328389	+0.019769	0.310880	+0.002260
Seismic bumps	TabICL	0.245852	0.264656	+0.018804	0.246950	+0.001098
Seismic bumps	TabPFN	0.248281	0.265708	+0.017427	0.249155	+0.000874
Splice	TabICL	0.769764	0.767873	-0.001891	0.771551	+0.001787
Splice	TabPFN	0.423380	0.447047	+0.023667	0.427804	+0.004423
Student dropout (46960)	TabICL	0.762596	0.781238	+0.018641	0.761666	-0.000931
Student dropout (46960)	TabPFN	0.714497	0.732177	+0.017679	0.712214	-0.002284

Post-hoc small-cluster sensitivity (Bank-16).

Comparator	Pooled mean	t-based 95% CI	LODO mean range	LODO sign reversal
CRUMB	$+0.010818$	$[+0.004686, +0.016951]$	$[+0.008908, +0.012196]$	No
Same-bank averaging	$+0.000365$	$[-0.000487, +0.001217]$	$[+0.000116, +0.000544]$	No

E.4 Original Bank-Size Robustness

Figure 7 reports the original formal-test results for the prespecified B4, B8, B16, and B32 banks. Results pooled across backbones remain secondary.

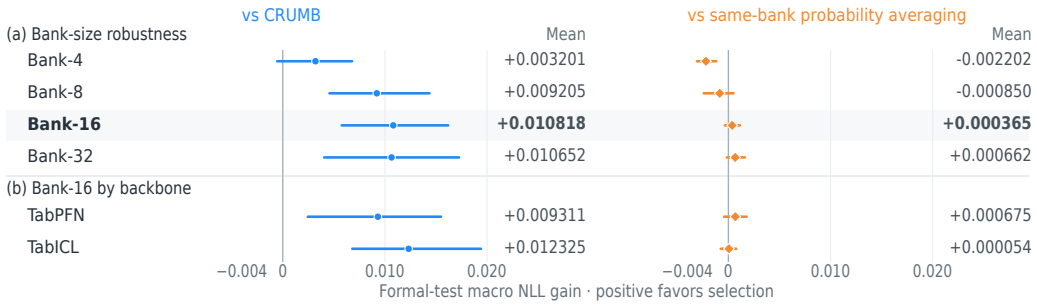


Figure 7: Original formal-test PC gains across prespecified banks and at B16 by backbone. Original paired dataset-cluster bootstrap intervals; positive gain favors PC. Panel (a) pools backbones secondarily; panel (b) separates them at B16. The columns show PC versus CRUMB and PC versus same-bank probability averaging.

Table 35: Original formal-test results for the prespecified bank sizes. Intervals come from the original analysis. Pooled W/T/L counts refer to dataset–model comparisons, not independent dataset clusters.

Backbone	Bank	Comparator	Gain	Original 95% interval	W/T/L
TabPFN	B4	CRUMB	+0.003847	[-0.001209, +0.008509]	7/0/5
TabPFN	B4	Averaging	-0.000449	[-0.002254, +0.001281]	6/0/6
TabPFN	B8	CRUMB	+0.008695	[+0.002736, +0.014315]	9/0/3
TabPFN	B8	Averaging	-0.000340	[-0.002150, +0.001180]	6/0/6
TabPFN	B32	CRUMB	+0.008562	[-0.001238, +0.016545]	10/0/2
TabPFN	B32	Averaging	+0.000869	[-0.000274, +0.002227]	8/0/4
TabICL	B4	CRUMB	+0.002554	[-0.000689, +0.006233]	9/0/3
TabICL	B4	Averaging	-0.003955	[-0.005921, -0.002051]	2/0/10
TabICL	B8	CRUMB	+0.009714	[+0.005307, +0.016152]	12/0/0
TabICL	B8	Averaging	-0.001361	[-0.002873, -0.000015]	5/0/7
TabICL	B32	CRUMB	+0.012741	[+0.006566, +0.020390]	11/0/1
TabICL	B32	Averaging	+0.000455	[-0.000293, +0.001283]	7/0/5
Pooled secondary	B4	CRUMB	+0.003201	[-0.000545, +0.006783]	16/0/8
Pooled secondary	B4	Averaging	-0.002202	[-0.003082, -0.001185]	8/0/16
Pooled secondary	B8	CRUMB	+0.009205	[+0.004574, +0.014357]	21/0/3
Pooled secondary	B8	Averaging	-0.000850	[-0.002422, +0.000501]	11/0/13
Pooled secondary	B32	CRUMB	+0.010652	[+0.004053, +0.017262]	21/0/3
Pooled secondary	B32	Averaging	+0.000662	[-0.000158, +0.001617]	15/0/9

E.5 Fresh-Holdout Post-Bank Comparisons

For the fresh holdout, we apply the fixed PC rule, CRUMB, and same-bank probability averaging to the datasets and seeds in Appendix A.1. CRUMB uses the same construction as before, with the fresh-holdout unit seed. Matched query gains are averaged over seeds within dataset and then equally over the ten datasets. Table 36 reports pointwise paired dataset-bootstrap intervals, separately from the original formal-test intervals above. Backbone-specific inference is primary; secondary pooling follows Appendix B.5. Neither backbone has an interval establishing that PC improves on CRUMB here. Against averaging, the interval lies above zero for TabICL but not TabPFN.

Table 36: Fresh-holdout post-bank comparisons. Gain is comparator NLL minus PC NLL; pointwise 95% intervals use 10,000 paired dataset-bootstrap replicates. W/T/L counts dataset effects.

Backbone / comparator	Gain	95% interval	W/T/L
TabPFN / CRUMB	+0.000303	[−0.014014, +0.014541]	6/0/4
TabPFN / averaging	+0.005198	[−0.000301, +0.013560]	7/0/3
TabICL / CRUMB	−0.016070	[−0.048950, +0.011451]	4/0/6
TabICL / averaging	+0.003785	[+0.000219, +0.009491]	6/0/4
Secondary pooled-backbone results			
Pooled / CRUMB	−0.007883	[−0.029737, +0.011203]	5/0/5
Pooled / averaging	+0.004492	[+0.000197, +0.011437]	7/0/3

F Comparator Coverage

F.1 Comparator Definitions

The comparator coverage analysis evaluates PC against canonical slot 0, uniform-random expected selection, minimum-entropy selection, CRUMB, and probability averaging at B4/B8/B16/B32 for both backbones. Canonical slot 0 uses the designated bank identity and is not the development-selected fixed reference (BEST_FIXED). Minimum entropy selects using candidate predictive distributions after bank inference. CRUMB constructs one context using unlabeled query-batch information, whereas PC and probability averaging use the complete evaluated prediction bank.

Original formal CRUMB configuration. We used the same project-local CRUMB implementation throughout the original formal test; no official implementation had been verified by us before that test. Within each dataset–seed unit, we standardize the preprocessed inputs using the 128-row support pool, then cluster all 256 unlabeled MCAR15 queries into 20 groups with scikit-learn K-means (k-means++ initialization, 10 restarts, Lloyd updates, 300 iterations, tolerance 10^{-4}, and the unit's formal seed from Appendix A).

For each nonempty query group, the Gaussian kernel bandwidth is the median positive pairwise Euclidean distance in the concatenated support and query-group rows, bounded below by 10^{-6} and set to 1 if none is positive. Starting empty, selection uses the SINGLETON_MMD greedy score in Appendix A, replacing $k(x, q)$ with its average over the query group, until 64 distinct support rows are selected; score ties take the first support index.

Queries with the same ordered context share one model fit and are predicted in batches of at most 16. Only support labels are used for fitting. Query labels are unavailable during context construction and prediction.

Both expressions use the clipping convention defined in Appendix B.

For uniform-random selection, we report the expected NLL, computed by averaging candidate NLLs:

$$B^{-1} \sum_{c=1}^{B} -\log p_c(y \mid x).$$

Probability averaging instead combines predictive probabilities before NLL is computed:

$$-\log\left(B^{-1} \sum_{c=1}^{B} p_c(y \mid x)\right).$$

Random selection uses one candidate prediction; probability averaging creates a prediction by combining probabilities. Their losses differ, and expected random-selection loss does not describe the query-harm distribution from actual random choices.

F.2 Complete Original-Bank Comparisons

Table 39 reports every evaluated bank size and comparator for both backbones, including intervals crossing zero. At B16, gains over canonical slot 0, uniform-random expected selection, minimum

entropy, and CRUMB have positive intervals for both backbones; superiority over averaging is not established in this original condition. The displayed pointwise intervals come from the supplementary comparator analysis. Although B16 point estimates agree with the original results, these intervals do not replace the original formal intervals in Appendix E.

Table 37 documents why CRUMB was selected for the original formal test. Panels A and B of Table 38 distinguish fixed-bank selection from retrieval, learned-context, active-acquisition, stream, and transfer approaches. This comparison describes the information and actions available to each method; it does not rank the broader families.

Table 37: Development-only selection of the one-context comparator. The metric is macro mean NLL: three seeds are averaged within each of 18 development datasets and datasets are weighted equally. The panel contains all ten eligible one-context methods considered before formal-test evaluation. CRUMB has the lowest development NLL for both models and was selected without using formal-test outcomes.

	TabPFN			TabICL		
Comparator	Dev. NLL	Rank	Selected	Dev. NLL	Rank	Selected
CRUMB	0.578118	1	yes	0.593944	1	yes
Class-balanced KNN	0.578724	2	–	0.603627	4	–
Singleton MMD greedy	0.581761	3	–	0.594557	2	–
Standardized KNN	0.587824	4	–	0.602252	3	–
Raw KNN / LocalPFN retrieval	0.591639	5	–	0.606921	5	–
Global coverage medoid	0.604045	6	–	0.627387	6	–
Stratified random	0.608827	7	–	0.630972	8	–
Canonical slot 0	0.611847	8	–	0.630655	7	–
Local farthest-first	0.643103	9	–	0.676962	9	–
MixturePFN approx. (inference-only)	0.690765	10	–	0.704482	10	–

Table 38: Panel A: How the method families differ in the unit of selection, available information, and contexts they can construct. This is a comparison of design choices, not a performance ranking. Our primary single-call evaluation chooses from a fixed bank of 16 contexts and permits fallback to the development-selected fixed reference. The K = 64 bank-size analysis extends the bank to 32 candidates.

Work family and examples	Selection unit	Information available	Candidate or action space	Primary objective
Local or latent retrieval (Thomas et al., 2024; Koshil et al., 2024)	Query	Support labels and query features before outcome	Free query-specific subset in raw or learned geometry	Localization and scalable prediction
Global learned or validation-optimized context (Feuer et al., 2024; Chen et al., 2026)	Dataset	Training data and labeled validation outcomes	Learned soft context or optimized sample and feature subset	Dataset-level predictive performance
Test-cluster distribution matching (Heredge et al., 2026)	Query cluster	Labeled support and unlabeled test-query batch	Free cluster-specific subset selected by distribution matching	Shift-aware batching and inference efficiency
Active acquisition (Treerath & Pittorino, 2026)	Acquisition round	Pool features and labels acquired in earlier rounds	Growing labeled context drawn from an acquisition pool	Label efficiency and predictive performance
Stream context management (Lee et al., 2026a)	Future stream query	Past labeled stream; current label affects only future memory	Dynamic bounded memory with admission and eviction	Prequential prediction under stream shift

Continued on next page

Table 38 – continued from previous page

Work family and examples	Selection unit	Information available	Candidate or action space	Primary objective
Source-target transfer (Lin & Li, 2026)	Target task and selected predictor	Target covariates plus labeled calibration and validation data	Transport-selected source anchors and validation-selected predictors	Transfer performance with average-risk safeguard
Current single-call diagnostic evaluation	Query observation	Labeled support, unlabeled query, and fixed pre-outcome signals	Fixed Bank-16 policies plus fixed-baseline fallback	Evaluate Oracle interpretation, identification, utility, and harm endpoints
Original formal post-bank evaluation	Query observation	Predictive distributions from a fixed candidate bank; no query labels or losses	Select one existing context at Bank-4/8/16/32; compare with same-bank probability averaging	Evaluate higher-information selection and same-bank aggregation

Table 38: Panel B: Which Oracle, null-model, and query-level harm questions are covered by the evaluations summarized in Panel A. Statements that an analysis was not identified refer to the reviewed primary evaluations, not to every paper in the method family.

Work family and examples	Oracle or null analysis	Query or tail-harm analysis
Local or latent retrieval (Thomas et al., 2024; Koshil et al., 2024)	No corresponding primary fixed-bank Raw Oracle or dependence-aware null analysis identified	Outside primary evaluation scope
Global learned or validation-optimized context (Feuer et al., 2024; Chen et al., 2026)	Validation optimization; no corresponding primary bank-null analysis identified	Outside primary evaluation scope
Test-cluster distribution matching (Heredge et al., 2026)	No corresponding primary Raw Oracle or dependence-aware null analysis identified	Outside primary evaluation scope
Active acquisition (Treerath & Pittorino, 2026)	No corresponding primary best-of-bank Oracle or null analysis identified	Outside primary evaluation scope
Stream context management (Lee et al., 2026a)	No corresponding primary best-of-bank Oracle or null analysis identified	No corresponding primary same-query harm or tail analysis identified
Source-target transfer (Lin & Li, 2026)	Theoretical oracle inequality; not Raw Oracle or a bank null	No-negative-transfer bound; not empirical same-query tails
Current single-call diagnostic evaluation	Development null diagnostics and score-level simulation checks	Same-query harm, $h > 0.05$ harm, P95, CVaR90, and paired utility
Original formal post-bank evaluation	Does not identify the Raw Oracle or qualify a bank null	Formal-test Bank-16 harm versus CRUMB plus paired mean utility and compute

Table 39: PC compared with all evaluated alternatives and original bank sizes. These supplementary pointwise 95% intervals are reported separately from the original formal intervals.

Backbone	Bank	Comparator	Gain	95% interval
TabPFN	B4	Averaging	-0.000449	[-0.002236, +0.001277]
TabPFN	B4	Slot 0	+0.025419	[+0.013144, +0.039241]
TabPFN	B4	CRUMB	+0.003847	[-0.000949, +0.008408]

Backbone	Bank	Comparator	Gain	95% interval
TabPFN	B4	Min. entropy	+0.029908	[+0.018691, +0.041012]
TabPFN	B4	UNIFORM_RANDOM	+0.018730	[+0.013522, +0.024452]
TabPFN	B8	Averaging	-0.000340	[-0.002105, +0.001192]
TabPFN	B8	Slot 0	+0.030268	[+0.016959, +0.044625]
TabPFN	B8	CRUMB	+0.008695	[+0.002940, +0.014443]
TabPFN	B8	Min. entropy	+0.047381	[+0.030730, +0.064184]
TabPFN	B8	UNIFORM_RANDOM	+0.020895	[+0.016126, +0.026449]
TabPFN	B16	Averaging	+0.000675	[-0.000450, +0.001802]
TabPFN	B16	Slot 0	+0.030884	[+0.018179, +0.044349]
TabPFN	B16	CRUMB	+0.009311	[+0.002250, +0.015485]
TabPFN	B16	Min. entropy	+0.065651	[+0.044756, +0.086452]
TabPFN	B16	UNIFORM_RANDOM	+0.023109	[+0.016557, +0.030288]
TabPFN	B32	Averaging	+0.000869	[-0.000232, +0.002199]
TabPFN	B32	Slot 0	+0.030134	[+0.017937, +0.042592]
TabPFN	B32	CRUMB	+0.008562	[-0.000958, +0.016327]
TabPFN	B32	Min. entropy	+0.083350	[+0.061319, +0.104863]
TabPFN	B32	UNIFORM_RANDOM	+0.023583	[+0.016431, +0.031312]
TabICL	B4	Averaging	-0.003955	[-0.005901, -0.002086]
TabICL	B4	Slot 0	+0.028734	[+0.015261, +0.044083]
TabICL	B4	CRUMB	+0.002554	[-0.000790, +0.006209]
TabICL	B4	Min. entropy	+0.030326	[+0.013531, +0.049077]
TabICL	B4	UNIFORM_RANDOM	+0.016740	[+0.010617, +0.023868]
TabICL	B8	Averaging	-0.001361	[-0.002884, -0.000012]
TabICL	B8	Slot 0	+0.035894	[+0.019188, +0.055564]
TabICL	B8	CRUMB	+0.009714	[+0.005268, +0.016173]
TabICL	B8	Min. entropy	+0.059805	[+0.032828, +0.085403]
TabICL	B8	UNIFORM_RANDOM	+0.023434	[+0.016538, +0.031511]
TabICL	B16	Averaging	+0.000054	[-0.000764, +0.000759]
TabICL	B16	Slot 0	+0.038505	[+0.022176, +0.058979]
TabICL	B16	CRUMB	+0.012325	[+0.006843, +0.019248]
TabICL	B16	Min. entropy	+0.085061	[+0.051027, +0.118033]
TabICL	B16	UNIFORM_RANDOM	+0.028053	[+0.019465, +0.037971]
TabICL	B32	Averaging	+0.000455	[-0.000291, +0.001294]
TabICL	B32	Slot 0	+0.038921	[+0.021101, +0.060198]
TabICL	B32	CRUMB	+0.012741	[+0.006496, +0.020284]
TabICL	B32	Min. entropy	+0.107850	[+0.066186, +0.147769]
TabICL	B32	UNIFORM_RANDOM	+0.029683	[+0.020024, +0.040476]
Pooled secondary	B4	Averaging	-0.002202	[-0.003080, -0.001162]
Pooled secondary	B4	Slot 0	+0.027077	[+0.014754, +0.040217]
Pooled secondary	B4	CRUMB	+0.003201	[-0.000573, +0.006838]
Pooled secondary	B4	Min. entropy	+0.030117	[+0.017725, +0.043091]
Pooled secondary	B4	UNIFORM_RANDOM	+0.017735	[+0.012161, +0.023846]
Pooled secondary	B8	Averaging	-0.000850	[-0.002437, +0.000518]
Pooled secondary	B8	Slot 0	+0.033081	[+0.018641, +0.049065]
Pooled secondary	B8	CRUMB	+0.009205	[+0.004631, +0.014248]
Pooled secondary	B8	Min. entropy	+0.053593	[+0.035943, +0.070099]
Pooled secondary	B8	UNIFORM_RANDOM	+0.022165	[+0.016808, +0.027756]
Pooled secondary	B16	Averaging	+0.000365	[-0.000337, +0.001111]
Pooled secondary	B16	Slot 0	+0.034694	[+0.020670, +0.050127]
Pooled secondary	B16	CRUMB	+0.010818	[+0.005921, +0.016291]
Pooled secondary	B16	Min. entropy	+0.075356	[+0.051822, +0.097418]
Pooled secondary	B16	UNIFORM_RANDOM	+0.025581	[+0.018705, +0.033188]
Pooled secondary	B32	Averaging	+0.000662	[-0.000155, +0.001660]
Pooled secondary	B32	Slot 0	+0.034528	[+0.020487, +0.050511]
Pooled secondary	B32	CRUMB	+0.010652	[+0.004195, +0.017190]
Pooled secondary	B32	Min. entropy	+0.095600	[+0.068206, +0.120636]
Pooled secondary	B32	UNIFORM_RANDOM	+0.026633	[+0.018550, +0.035526]

G CANDIDATE-BANK AND OPERATING-CONDITION SENSITIVITIES

We designed these analyses after examining the original formal outcomes. They reuse the same 12 datasets to study how PC's gains change with the input condition and candidate bank. All intervals are supplementary and pointwise. Each paired comparison uses predictions from the same inference environment.

G.1 INPUT-CONDITION SENSITIVITY

NO_ADDED_MCAR removes only the added MCAR15 mask and retains native missingness. At B16, PC gains over CRUMB are +0.007894 [+0.000898, +0.012791] for TabPFN and +0.014885 [+0.007520, +0.022089] for TabICL. Gains over averaging are +0.001856 [+0.000295, +0.003852] and +0.000105 [−0.000802, +0.000931], respectively.

When switching from MCAR15 to NO_ADDED_MCAR in the same environment, the change in PC gain over averaging is +0.001181 [+0.000380, +0.002313] for TabPFN; the corresponding TabICL interval crosses zero. For both backbones, the interval for the change in PC's gain over CRUMB crosses zero. Table 40 compares methods within each input condition; Table 41 compares their gains before and after removing the mask. A positive interval for one backbone and a crossing interval for the other do not by themselves establish a difference between backbones.

Table 40: PC gains under MCAR15 and without the added mask. The reference and modified settings use predictions from the same supplementary inference environment. Intervals are supplementary, pointwise 95% intervals.

Backbone	Setting	Comparator	Gain	95% interval
TabPFN	MCAR15	Averaging	+0.000675	[-0.000472, +0.001789]
TabPFN	MCAR15	CRUMB	+0.009311	[+0.002155, +0.015537]
TabPFN	NO_ADDED_MCAR	Averaging	+0.001856	[+0.000295, +0.003852]
TabPFN	NO_ADDED_MCAR	CRUMB	+0.007894	[+0.000898, +0.012791]
TabICL	MCAR15	Averaging	+0.000054	[-0.000751, +0.000745]
TabICL	MCAR15	CRUMB	+0.012325	[+0.006966, +0.019259]
TabICL	NO_ADDED_MCAR	Averaging	+0.000105	[-0.000802, +0.000931]
TabICL	NO_ADDED_MCAR	CRUMB	+0.014885	[+0.007520, +0.022089]
Pooled secondary	MCAR15	Averaging	+0.000365	[-0.000343, +0.001119]
Pooled secondary	MCAR15	CRUMB	+0.010818	[+0.005830, +0.016192]
Pooled secondary	NO_ADDED_MCAR	Averaging	+0.000980	[-0.000167, +0.002353]
Pooled secondary	NO_ADDED_MCAR	CRUMB	+0.011390	[+0.006435, +0.015999]

Table 41: Change in PC's gain over the named comparison method when moving from the reference to the modified setting. Positive values mean a larger relative gain. Intervals are pointwise 95% intervals.

Backbone	Change		Comparator	Change in gain	95% interval
TabPFN	No-added MCAR15	minus	Averaging	+0.001181	[+0.000380, +0.002313]
TabPFN	No-added MCAR15	minus	CRUMB	-0.001417	[-0.005811, +0.003394]
TabICL	No-added MCAR15	minus	Averaging	+0.000050	[-0.000853, +0.000973]
TabICL	No-added MCAR15	minus	CRUMB	+0.002559	[-0.001505, +0.006328]

G.2 CANDIDATE-BANK CONSTRUCTION SENSITIVITY

The original B16 bank contains nine non-random and seven stratified-random candidates. Identity-resampled banks keep these counts while changing which candidate identities are selected. High-random banks instead use two non-random and fourteen stratified-random candidates. We evaluate the original bank and ten repetitions of each alternative construction. PC and averaging are recomputed for each subset, and subset repetitions are averaged within dataset before bootstrap resampling. A comparison with canonical slot 0 is included only when the subset contains that slot.

For TabICL, the high-random-minus-original change in comparative gain is −0.000844 [−0.001656, −0.000167] against averaging and −0.002510 [−0.004708, −0.000601] against CRUMB. The negative changes indicate smaller gains in the evaluated high-random construction. Its absolute gain against averaging is −0.000789 [−0.002089, +0.000415], which does not establish inferiority; its gain against CRUMB remains +0.009815 [+0.003808, +0.016838].

For TabPFN, high-random absolute gains are +0.000455 [−0.000809, +0.001809] against averaging and +0.006354 [−0.004769, +0.014609] against CRUMB. Its paired construction-change intervals cross zero. Identity-resampling changes also have crossing intervals for both backbones. Construction counts and selection frequencies describe the evaluated subsets; they do not isolate a causal mechanism or establish invariance to composition.

Table 42: PC gains for each candidate-bank construction, relative to averaging and CRUMB. Intervals are supplementary, pointwise 95% intervals.

Backbone	Setting	Comparator	Gain	95% interval
TabPFN	High-random	Averaging	+0.000455	[-0.000809, +0.001809]
TabPFN	High-random	CRUMB	+0.006354	[-0.004769, +0.014609]
TabPFN	Identity-resampled	Averaging	+0.000824	[-0.000050, +0.001755]
TabPFN	Identity-resampled	CRUMB	+0.009630	[+0.001605, +0.016514]
TabPFN	Original	Averaging	+0.000675	[-0.000458, +0.001791]
TabPFN	Original	CRUMB	+0.009311	[+0.002160, +0.015516]
TabICL	High-random	Averaging	-0.000789	[-0.002089, +0.000415]
TabICL	High-random	CRUMB	+0.009815	[+0.003808, +0.016838]
TabICL	Identity-resampled	Averaging	-0.000328	[-0.001230, +0.000502]
TabICL	Identity-resampled	CRUMB	+0.012066	[+0.006702, +0.018889]
TabICL	Original	Averaging	+0.000054	[-0.000757, +0.000779]
TabICL	Original	CRUMB	+0.012325	[+0.006923, +0.019311]
Pooled secondary	High-random	Averaging	-0.000167	[-0.001389, +0.001040]
Pooled secondary	High-random	CRUMB	+0.008085	[+0.000826, +0.014754]
Pooled secondary	Identity-resampled	Averaging	+0.000248	[-0.000499, +0.001006]
Pooled secondary	Identity-resampled	CRUMB	+0.010848	[+0.005380, +0.016544]
Pooled secondary	Original	Averaging	+0.000365	[-0.000346, +0.001114]
Pooled secondary	Original	CRUMB	+0.010818	[+0.005859, +0.016310]

Table 43: Change in PC's gain over the named comparison method when moving from the reference to the modified setting. Positive values mean a larger relative gain. Intervals are pointwise 95% intervals.

Backbone	Change	Comparator	Change in gain	95% interval
TabPFN	Identity minus original	Averaging	+0.000149	[-0.000570, +0.000790]
TabPFN	High-random minus original	Averaging	-0.000220	[-0.001188, +0.000447]
TabPFN	Identity minus original	CRUMB	+0.000319	[-0.000896, +0.001525]
TabPFN	High-random minus original	CRUMB	-0.002957	[-0.007619, +0.000460]

Backbone	Change	Comparator	Change in gain	95% interval
TabICL	Identity minus original	Averaging	-0.000383	[-0.000757, +0.000068]
TabICL	High-random minus original	Averaging	-0.000844	[-0.001656, -0.000167]
TabICL	Identity minus original	CRUMB	-0.000260	[-0.001102, +0.000835]
TabICL	High-random minus original	CRUMB	-0.002510	[-0.004708, -0.000601]

G.3 BANK-SIZE CHARACTERIZATION

We report comparisons between methods at each of B16/B32/B64, separately from comparisons of the same method before and after expansion from B32 to B64. At B64, PC-versus-averaging gain is +0.001288 [+0.000332, +0.002316] for TabPFN and +0.000112 [−0.000670, +0.000888] for TabICL.

All four direct-expansion intervals in Table 45 cross zero. For TabICL, the change in PC's gain over averaging from B32 to B64 is −0.000343 [−0.000701, −0.000033]; this relative change does not show that either method's own loss improves. The complete method-contrast table also retains the TabPFN B64-versus-CRUMB interval crossing zero. Added B64 candidates are stratified-random, so bank size and construction are not causally isolated. These comparisons do not establish a general scaling law.

Table 44: PC gains at each bank size, using predictions from the same supplementary inference environment. Intervals are supplementary, pointwise 95% intervals.

Backbone	Setting	Comparator	Gain	95% interval
TabPFN	B4	Averaging	-0.000449	[-0.002236, +0.001292]
TabPFN	B4	CRUMB	+0.003847	[-0.001063, +0.008545]
TabPFN	B8	Averaging	-0.000340	[-0.002067, +0.001215]
TabPFN	B8	CRUMB	+0.008695	[+0.002903, +0.014452]
TabPFN	B16	Averaging	+0.000675	[-0.000472, +0.001789]
TabPFN	B16	CRUMB	+0.009311	[+0.002155, +0.015537]
TabPFN	B32	Averaging	+0.000869	[-0.000240, +0.002190]
TabPFN	B32	CRUMB	+0.008562	[-0.001184, +0.016354]
TabPFN	B64	Averaging	+0.001288	[+0.000332, +0.002316]
TabPFN	B64	CRUMB	+0.008808	[-0.002321, +0.017354]
TabICL	B4	Averaging	-0.003955	[-0.005909, -0.002096]
TabICL	B4	CRUMB	+0.002554	[-0.000804, +0.006226]
TabICL	B8	Averaging	-0.001361	[-0.002865, -0.000015]
TabICL	B8	CRUMB	+0.009714	[+0.005323, +0.015899]
TabICL	B16	Averaging	+0.000054	[-0.000751, +0.000745]
TabICL	B16	CRUMB	+0.012325	[+0.006966, +0.019259]
TabICL	B32	Averaging	+0.000455	[-0.000299, +0.001276]
TabICL	B32	CRUMB	+0.012741	[+0.006514, +0.020183]
TabICL	B64	Averaging	+0.000112	[-0.000670, +0.000888]
TabICL	B64	CRUMB	+0.013089	[+0.006589, +0.021217]
Pooled secondary	B4	Averaging	-0.002202	[-0.003080, -0.001171]
Pooled secondary	B4	CRUMB	+0.003201	[-0.000598, +0.006723]
Pooled secondary	B8	Averaging	-0.000850	[-0.002416, +0.000485]
Pooled secondary	B8	CRUMB	+0.009205	[+0.004600, +0.014324]
Pooled secondary	B16	Averaging	+0.000365	[-0.000343, +0.001119]
Pooled secondary	B16	CRUMB	+0.010818	[+0.005830, +0.016192]
Pooled secondary	B32	Averaging	+0.000662	[-0.000158, +0.001628]
Pooled secondary	B32	CRUMB	+0.010652	[+0.004084, +0.017243]
Pooled secondary	B64	Averaging	+0.000700	[-0.000039, +0.001473]
Pooled secondary	B64	CRUMB	+0.010948	[+0.003108, +0.018325]

Table 45: Change in each method's NLL when the bank expands from 32 to 64 candidates. Values are B32 NLL minus B64 NLL, so positive values favor the larger bank. Both predictions come from the same supplementary inference environment.

Backbone	Method	Gain	95% interval
TabPFN	PC	+0.000245	[-0.001295, +0.001545]
TabPFN	Averaging	-0.000174	[-0.002062, +0.001106]
TabICL	PC	+0.000348	[-0.001068, +0.001398]
TabICL	Averaging	+0.000691	[-0.000895, +0.001952]

Table 46: Change in PC's gain over the named comparison method when moving from the reference to the modified setting. Positive values mean a larger relative gain. Intervals are pointwise 95% intervals.

Backbone	Change	Comparator	Change in gain	95% interval
TabPFN	B64-B32	Averaging	+0.000419	[-0.000097, +0.000918]
TabPFN	B64-B32	CRUMB	+0.000245	[-0.001332, +0.001542]
TabICL	B64-B32	Averaging	-0.000343	[-0.000701, -0.000033]
TabICL	B64-B32	CRUMB	+0.000348	[-0.001064, +0.001402]

H Alternative Use of the Same Support Pool

H.1 Different Context Budgets

Full support puts all 128 labeled support examples into one context, giving B1/K128. PC and averaging use 16 contexts with 64 examples each, while CRUMB uses K64. All methods have access to the same labeled support pool. Full support uses more of that pool in a single context; it does not receive additional labeled data.

H.2 Complete Practical Comparisons

The complete comparison includes both backbones, MCAR15 and NO_ADDED_MCAR, and PC, averaging, and CRUMB as comparators. Gain is comparator NLL minus full-support NLL, so positive values favor full support. Results against PC appear in Table 47.

Four of the 12 rows compare full support with PC. All four intervals for gains over CRUMB are positive. Against averaging, the TabPFN NO_ADDED_MCAR interval is positive and the other three cross zero. Full support is therefore a relevant practical alternative, with a positive pointwise interval in one evaluated comparison against PC. Since full support uses a larger context, the comparison does not hold K fixed or establish a runtime advantage.

Table 47: Using the same 128 labeled support examples in different ways. Full support puts all examples in one context (B1/K128); PC and averaging use 16 contexts of 64 examples (B16/K64), and CRUMB uses K64. Positive gain means lower NLL for full support. Intervals are pointwise 95% intervals.

Backbone	Condition	Comparator	Gain	95% interval
TabPFN	MCAR15	CRUMB	+0.014614	[+0.003003, +0.029163]
TabPFN	MCAR15	PC	+0.005302	[-0.002431, +0.015642]
TabPFN	MCAR15	Averaging	+0.005978	[-0.002341, +0.017285]
TabPFN	No added MCAR	CRUMB	+0.017794	[+0.006440, +0.033100]

Backbone	Condition	Comparator	Gain	95% interval
TabPFN	No added MCAR	PC	+0.009900	[+0.000062, +0.023321]
TabPFN	No added MCAR	Averaging	+0.011756	[+0.001026, +0.027057]
TabICL	MCAR15	CRUMB	+0.023336	[+0.006676, +0.049453]
TabICL	MCAR15	PC	+0.011011	[-0.006842, +0.039005]
TabICL	MCAR15	Averaging	+0.011065	[-0.007318, +0.039422]
TabICL	No added MCAR	CRUMB	+0.030973	[+0.012417, +0.060065]
TabICL	No added MCAR	PC	+0.016088	[-0.004737, +0.050326]
TabICL	No added MCAR	Averaging	+0.016193	[-0.004694, +0.050811]

I Query-Wise Reliability

I.1 Backbone-Specific Diagnostics

Reliability summaries report $h = L_{\mathrm{PC}} - L_{\mathrm{comparator}}$, so positive harm means that PC increases loss on a query. The primary table reports each backbone separately against CRUMB and averaging for original B16 MCAR15, B16 NO_ADDED_MCAR, and B64 MCAR15. We report how often PC increases loss, how often the increase exceeds 0.05, the 95th percentile of harm, and the mean harm at or above the 90th percentile. The backbone-specific summaries of the original B16 predictions were computed in the later supplementary analysis.

I.2 Pooling and Secondary Summaries

For each setting, backbone, and comparator, we combine matched query harms across datasets and seeds before computing the quantile and tail mean. The secondary pooled table additionally pools both backbones. If $q_{0.90}$ denotes the empirical 90th percentile of that pool, the tail mean averages all harms satisfying $h \geq q_{0.90}$, including ties. We do not average separately computed cellwise quantiles or tail means.

The single-call CVaR90 reported in Appendix D instead averages the largest $\lceil 0.10n \rceil$ harms within its specified calculation. The two tail summaries use different estimators, as defined in Appendix B.4.

For original B16, secondary pooled harm against CRUMB is 54.71%, with 21.11% exceeding 0.05, P95 of 0.232130, and inclusive q90 tail mean of 0.284615. Against averaging, the corresponding summaries are 28.61%, 5.43%, 0.054480, and 0.079978.

I.3 Interpretation

These measures describe how often PC raises query loss and the size of those increases. The threshold of 0.05 is an operational choice. Queries do not count as additional independent datasets, and a lower observed harm rate in one setting is not a guarantee of improvement elsewhere. A method can reduce average loss while increasing loss on a majority of queries. These summaries do not establish application-specific safety or provide a complete assessment of calibration.

Table 48: Query-level loss increases for each model and setting. The original B16 predictions are summarized in a later supplementary analysis. Rates are proportions; P95 and the mean harm at or above the 90th percentile are in NLL units.

Backbone	Setting	Comparator	P(h>0)	P(h>0.05)	P95	q90 tail
TabPFN	B16 MCAR15	Averaging	0.280382	0.050890	0.050681	0.074796
TabPFN	B16 MCAR15	CRUMB	0.548069	0.215929	0.228032	0.284471
TabPFN	B64 MCAR15	Averaging	0.185221	0.034614	0.037508	0.050830
TabPFN	B64 MCAR15	CRUMB	0.547309	0.221788	0.240402	0.300171
TabPFN	B16 no added	Averaging	0.269531	0.045356	0.043915	0.062649
TabPFN	B16 no added	CRUMB	0.537760	0.204753	0.204991	0.250381

Backbone	Setting	Comparator	P(h>0)	P(h>0.05)	P95	q90 tail
TabICL	B16 MCAR15	Averaging	0.291775	0.057726	0.057386	0.084847
TabICL	B16 MCAR15	CRUMB	0.546115	0.206272	0.237399	0.284657
TabICL	B64 MCAR15	Averaging	0.207465	0.041341	0.044127	0.065270
TabICL	B64 MCAR15	CRUMB	0.552192	0.217556	0.242964	0.297621
TabICL	B16 no added	Averaging	0.279080	0.060221	0.060224	0.090485
TabICL	B16 no added	CRUMB	0.556424	0.204427	0.218990	0.267097

Table 49: Secondary harm summaries pooling both models. Matched query harms are combined before computing quantiles and the tail mean. Rates are proportions; P95 and tail mean are in NLL units.

Backbone	Setting	Comparator	P(h>0)	P(h>0.05)	P95	q90 tail
Pooled secondary	B16 MCAR15	Averaging	0.286079	0.054308	0.054480	0.079978
Pooled secondary	B16 MCAR15	CRUMB	0.547092	0.211100	0.232130	0.284615
Pooled secondary	B64 MCAR15	Averaging	0.196343	0.037977	0.041400	0.058242
Pooled secondary	B64 MCAR15	CRUMB	0.549750	0.219672	0.241677	0.299124
Pooled secondary	B16 no added	Averaging	0.274306	0.052789	0.052715	0.076810
Pooled secondary	B16 no added	CRUMB	0.547092	0.204590	0.209542	0.258716

I.4 Fresh-Holdout Query-Wise Harm

Table 50 reports how often PC increases query loss on the fresh holdout. Each backbone contributes 5,120 matched queries across ten datasets and two seeds. The rates pool these observations; they are descriptive summaries, not tests using queries as independent dataset replicates. Secondary pooled rates combine the two backbones with equal weight.

Table 50: Percentage of fresh-holdout queries on which PC increases loss relative to each comparison method. Harm is PC NLL minus comparator NLL.

Backbone	Comparator	$h > 0$	$h > 0.05$
TabPFN	CRUMB	58.40%	33.93%
TabICL	CRUMB	59.30%	33.87%
TabPFN	Same-bank averaging	31.21%	8.75%
TabICL	Same-bank averaging	33.77%	10.33%
Pooled (secondary)	CRUMB	58.85%	33.90%
Pooled (secondary)	Same-bank averaging	32.49%	9.54%

J Computational Cost and Implementation Scope

J.1 Original Measured Costs

Under the original implementation, the Bank-16 post-bank evaluation averaged 2,679.861 physical predict calls and 797.949530 seconds, compared with 26.917 calls and 7.729405 seconds for one-context CRUMB. The reported ratios are 99.6-fold for calls and 103.2-fold for wall-clock time. For calls and time, we average the three seeds within each dataset–backbone cell, then average the 24 cells. The resulting mean number of calls need not be an integer. Maximum recorded peak VRAM was 198.6 MiB for Bank-16 and 197.2 MiB for CRUMB.

These timings were measured in the original formal test, not on the fresh holdout. Runtime measurements were obtained on one 32 GB vGPU with 12 Intel Xeon Platinum 8352V vCPUs and 62 GB RAM, using PyTorch 2.5.1, Python 3.12, CUDA 12.4, and Ubuntu 22.04.

Bank time and call counts sum the candidate prediction runs, while peak memory takes their maximum. The analysis uses each run's recorded wall-clock duration, or fitting plus prediction-and-alignment time when a combined duration is unavailable. For memory it uses the recorded peak, falling back to an incremental peak only when the former is unavailable.

J.2 Inference Paths and Limits of Extrapolation

The measured ratios depend on the inference implementation. The TabPFN candidate-prediction path used a batch size of one, while one-context prediction could batch queries that shared a context. The call ratio therefore reflects these execution choices as well as the candidate bank; it is not an intrinsic complexity ratio. PC and probability averaging use the same bank predictions, but their selection and aggregation steps need not have identical total cost.

The supplementary runs lack controlled timing on matched hardware. They therefore provide no controlled estimate of B64 speed ratios, full-support runtime relative to K64 methods, or worker speedups.

Table 51: Measured cost of evaluating Bank-16 relative to one-context CRUMB in our original implementation. Calls and time are averaged over three seeds within each of the 24 dataset–model combinations, then over those combinations. Average call counts can therefore be non-integer. Peak VRAM is the maximum across combinations. The absolute values give ratios of 99.6 for prediction calls and 103.2 for wall-clock time.

Quantity	One-context CRUMB	Bank-16 post-bank	Ratio
Physical predict calls	26.917	2679.861	99.6×
Wall-clock (s)	7.729405	797.949530	103.2×
Peak VRAM (MiB)	197.2	198.6	1.01×

Table 52: Original formal-test PC performance and the cost of computing its candidate bank. NLL is primary; accuracy and macro one-vs-rest AUROC are secondary. Calls and time use the averaging described in Table 51, and peak VRAM is the maximum across combinations. Averaging reuses the bank predictions, whereas CRUMB constructs one context using an unlabeled query batch.

Bank	Selected NLL	Accuracy	Macro AUROC	Physical calls	Wall-clock (s)	Peak VRAM (MiB)
4	0.456496	0.815104	0.734430	777.5	225.4	196.2
8	0.450492	0.817980	0.742180	1591.9	475.4	198.6
16	0.448878	0.818034	0.746134	2679.9	797.9	198.6
32	0.449045	0.817708	0.745628	4855.9	1427.4	198.6