CERTIFIED CONTEXT INTEGRITY FOR IN-CONTEXT PREDICTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern in-context predictors treat a support set as part of the inference input: tabular foundation models condition on support tables, and large language models condition on demonstrations. This opens a poisoning surface that needs no retraining, because whoever controls the context can change predictions without touching the model. Against four tabular foundation models and 15-NN on six real tables, a median of one to eight inserted rows flips an undefended prediction, and for TabICL a single row suffices on half the tables. We introduce *certified context partitioning*, a retraining-free, black-box wrapper that issues a deterministic certificate for each query. Each support row is assigned to a shard by a cryptographic hash of its own features, the frozen predictor runs once on every realised shard, and the prediction is the plurality of the shard votes. An insertion, deletion or relabelling can change only one shard, so the vote gap certifies how many such operations the prediction survives; feature edits receive half that radius. Across 20 OpenML/UCI tables, 12 base learners, shard counts up to 256 and three seeds, a median certified radius of 7 rows ($m = 16$) costs a median 1.0 accuracy point. With every configuration chosen on held-out validation queries, partitioned in-context prediction beats certified k-NN, the closest retraining-free baseline, by 0.67, 1.00 and 0.94 points of certified accuracy at radii 1, 2 and 4 (winning on 14, 14 and 13 of 20 tables), while the same certificate over classical learners does not; under arbitrary edits the two are level unless rows carry trusted identifiers. Two white-box adaptive attacks that know the model, hash and partition produce no certificate violation in 3,600 attacks on 1,800 defended queries, and for TabICL, TabPFN v2.6 and Mitra v2 the median successful insertion attack needs exactly $r + 1$ rows, so the bound is attained. Content addressing costs no measurable accuracy against a balanced random partition, and partitioning cuts peak GPU memory 12–29$\times$ at 20,000 support rows. A learning-curve law predicts certified accuracy to a median error of 0.6 points and picks the best shard count outright in 60–74% of cells. Changing only the partitioner and backend certifies Qwen3.5-2B and Qwen3.5-4B on SST-2 at 89.0% and 88.0% against any eight inserted, deleted or relabelled demonstrations out of 512. Context supplied at inference time can therefore be treated as a certifiable security boundary, with deterministic guarantees and no retraining.

1 INTRODUCTION

Tabular foundation models such as TabPFN (Hollmann et al., 2023; 2025) and TabICL (Qu et al., 2025) do not train on your data. You hand them a table of labelled examples when you ask a question, they read it in one forward pass, and they answer. A language model that classifies from a few demonstrations works the same way. Because nothing has to be trained, these models are easy to deploy, and they now sit inside retrieval pipelines and shared services whose context is assembled from many sources. That convenience creates a new weak point. In classical data poisoning, an attacker corrupts a training set and must wait for the model to be retrained (Biggio et al., 2012; Steinhardt et al., 2017). Here there is nothing to wait for: the table handed over with the request *is* the model. A credit model that reads records pooled from partner lenders, or a ticket router whose demonstrations come from users, answers a party other than the one who wrote its context. Add a few rows, or relabel some that are already there, and the next answer changes.

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.

It takes very little. A simple white-box attack, which copies the query, attaches the label the attacker wants and inserts it, changes an undefended prediction of TabPFN v2, TabPFN v2.6, TabICL, Mitra v2 or 15-NN with a median of one to eight rows, on six real tables of up to 4,000 rows (15 attacked queries per table). For TabICL a single row is enough on `phoneme`, `satimage` and `spambase`.

Existing defences do not fit this setting. Defences that change how a model is trained (Levine & Feizi, 2021; Wang et al., 2022a) cannot be applied to a frozen model behind an API, and anything that must be recomputed for each table sits on the critical path of every request. The certificates that avoid retraining do so by replacing the model: they vote over nearest neighbours (Jia et al., 2022), compare distances to class prototypes in an embedding space (Wang et al., 2024), or handle short retrieved passages one at a time (Xiang et al., 2026). And a guarantee that holds on average over tables answers the wrong question: a deployer needs to know about *this* prediction, on *this* table, now.

Our starting point is that in-context models make a much simpler defence possible. Because such a model learns nothing ahead of time, we can split the context into pieces by hashing each row on its own content, run the frozen model once on each piece, and let the pieces vote. One poisoned row can change only one piece, so the voting margin tells us, for every single prediction, how many rows an attacker would have to insert, delete or relabel to change it. This is the voting argument that Deep Partition Aggregation (Levine & Feizi, 2021) uses for training sets, moved to inference time, where it needs no retraining, treats the model as a black box, and works unchanged for each new model release.

Contributions. (i) A per-prediction certificate for a context that changes with every request (§3, Figure 1). It covers r inserted, deleted or relabelled rows, $\lfloor r/2 \rfloor$ when features may also be edited, and r for every edit when rows carry trusted identifiers (Corollary 1). (ii) A large measurement of its price over 20 tables, 7 tabular foundation models, 5 classical learners and 3 seeds (§5): about one accuracy point for a median guarantee of seven rows. (iii) Evidence that the foundation model is what makes it work (§6). With the same certificate, in-context learners certify 0.6–1.4 points more than classical ones, which on held-out queries puts them ahead of certified k-NN at small radii (significant at radius 2 on every split), further ahead on larger tables (Appendix L), and ahead under arbitrary edits too once rows carry identifiers (Appendix F). (iv) Adaptive attacks that test how tight the guarantee is, including edits that change features and labels together (§7). (v) A learning-curve rule that predicts the certificate and picks the number of pieces before deployment (§8). (vi) The same certificate for LLMs that classify from demonstrations (§9). In the 20-table benchmark we report no certified accuracy above radius 24, where our grid is truncated (§4); Appendix L uses a wider grid.

2 Related work

Partition and bagging certificates. Deep Partition Aggregation (Levine & Feizi, 2021) hashes each *training* sample to one of k partitions, trains one model per partition, and certifies the plurality: a poisoned sample changes at most one base model, so the vote gap bounds the poison count. Finite Aggregation (Wang et al., 2022a) and Run-Off Election (Rezaei et al., 2023) tighten the bound, Chen et al. (2022) certify hash bagging collectively, and Jia et al. (2021) certify bagging intrinsically. Propositions 1–2 are DPA's argument with the training set replaced by a caller-supplied context: DPA already requires the hash to depend on the sample alone (our A1), hashes pixels only, sorts each partition by content and label (our A3), and keys its label-flip variant on the unlabelled sample so that a flip touches one partition, not two. What changes is where the certificate lives: DPA certifies a trained artefact once, whereas here nothing is fitted, the partition is rebuilt per request and the certificate is recomputed from that request's votes. That is practical only because an in-context learner needs no training and stays accurate on shards of a few dozen rows, which §5–8 measure. Hammoudeh & Lowd (2024) partition features rather than rows, which complements our feature-edit radius.

Retraining-free certificates over a support set. Jia et al. (2022) show that k-NN and radius-NN carry an intrinsic certificate: one poisoned row moves at most one neighbour vote, whatever the operation. Certified k-NN is our primary baseline, the one method that is retraining-free, per-query and

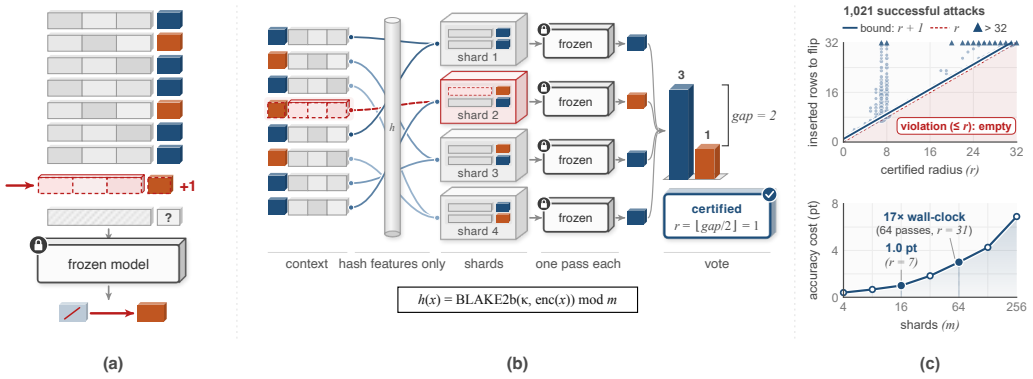


Figure 1: **Certified context partitioning.** **(a)** One inserted row flips a frozen model's prediction. **(b)** Rows are assigned to m shards by $h(x) = \mathrm{BLAKE2b}(\kappa, \mathrm{enc}(x)) \bmod m$, computed on features only. The frozen model runs once per shard and the shards vote. One row can change at most one vote: the poisoned row takes the tally from 4–0 (gap $= 4$, $r = 2$) to 3–1 (gap $= 2$, $r = 1$; ties go to blue, the lower class index), so the prediction survives and one more inserted row still cannot flip it. **(c)** Top: the 1,021 successful insertion attacks on TabPFN v2, v2.6 and TabICL at $m \in \{16, 64\}$ (§7); none needs r or fewer rows and 69.7% need exactly $r + 1$. Bottom: TabPFN v2 pays 1.0 accuracy point at $m = 16$ ($r = 7$); at $m = 64$ ($r = 31$) its 64 shard passes take 17× the undefended wall-clock.

deterministic on the same inputs. Its radius never exceeds $\lfloor k/2 \rfloor$ however large the table, whereas ours is $\lfloor m/2 \rfloor$ and m can be set in proportion to the expected table size; conversely it covers feature edits at full radius and we at half, so §6 compares under both threats. FCert (Wang et al., 2024) certifies few-shot foundation-model classification against poisoned support samples through robust distances to class prototypes in embedding space, and RobustRAG (Xiang et al., 2026) isolates retrieved passages and aggregates per-passage answers. Both certify a mutable inference-time context, as we do; both replace or bypass the model's joint in-context computation over the context, which we keep as a black box and which, on tables of thousands of rows, carries the accuracy. FCert needs a row-local embedding, and the tabular models here embed a row only jointly with its context; applied to raw query-standardised features, its trimmed class-distance rule reaches a median clean accuracy of 76.5% (seed 0), below the certified accuracy that both the partition and certified k-NN keep at radius 8.

Smoothing, ICL attacks, and sample efficiency. Randomised smoothing certifies test-time perturbations (Cohen et al., 2019) and extends to poisoning (Weber et al., 2023) and entity hierarchies (Scholten et al., 2023); those guarantees are probabilistic, ours is deterministic given the realised votes. In-context learning is vulnerable to poisoned demonstrations (Zhao et al., 2024; He et al., 2025), defences so far are empirical (Ren et al., 2025), and Djilani et al. (2025) study test-time attacks on TabPFN and TabICL. The Lethal Dose Conjecture (Wang et al., 2022b) relates a learner's sample requirement to the poison it tolerates; Proposition 3 makes a narrower, testable claim from the measured learning curve, and we report how well it holds.

3 Certified Context Partitioning

Setting and threat model. A support table S is a multiset of rows (x, y), $x \in \mathbb{R}^d$, $y \in \{0, \ldots, K-1\}$. A frozen in-context learner f maps a support table and query x_q to a class. The deployer receives S per request from a source it does not fully control. The adversary knows f, the partition scheme, the hash key if any, the table and the query, and may apply ρ operations from {insert a row, delete a row, change a label, change features}. We want a per-query r such that no r operations change the prediction. Only the support rows are untrusted: model weights, the query (it comes from the party receiving the prediction), hash key and any preprocessing frozen before the request are trusted, and the guarantee says nothing about poisoned checkpoints, schemas or queries. A certificate states *invariance*; certified accuracy counts a query only if it is also correct.

Content-addressed partitioning. Let $h(x) = \mathrm{BLAKE2b}(\kappa, \mathrm{enc}(x)) \bmod m$, with enc a canonical float32 byte encoding and κ an optional key, and $S_s = \{(x, y) \in S : h(x) = s\}$. The model runs once per realised shard, shard s casts the hard vote $v_s = f(S_s, x_q)$, and the prediction is the plurality over realised votes, ties to the smaller class index. The hash *reads only the row being hashed*, not its position, the other rows or per-run randomness, so an inserted row occupies one shard and moves nothing else; balance shard sizes, sort or assign by index, and this accounting collapses. It also *excludes the label*, so a relabelled row stays put and a label flip touches one shard, not two.

A1 (row-local, fixed assignment): h depends only on the row's canonical features and on m, κ, enc, all fixed before S is seen, so m may not depend on $|S|$ or S, and the design rule of §8 is run once on trusted data, such as a vetted snapshot of the same source. **A2** (realised shards): votes are counted over non-empty shards. **A3** (per-shard inference): the vote of shard s is a function of the multiset S_s and x_q alone, so preprocessing is fitted within the shard or frozen, and the vote ignores row order, which is not a counted operation. **A4** (canonical encoding): enc is deterministic across pipelines. **A5** (no cross-shard flow): batched evaluation of shards is vote-identical to sequential evaluation. **A6** (determinism): re-evaluating a shard on the deployed device returns the same vote. §4 says how each is enforced or measured.

The vote-gap certificate. Let n_c count realised shards voting c, $c_1 = \min \arg\max_c n_c$, and

$$\mathrm{gap} = \min_{c \neq c_1} \big(n_{c_1} - n_c - \mathbf{1}[c < c_1]\big), \qquad r = \lfloor \mathrm{gap}/2 \rfloor. \tag{1}$$

Equivalently $\mathrm{gap} = n_{c_1} - n_{c_2} - \mathbf{1}[c_2 < c_1]$ with c_2 the *smallest-index* highest-count rival; a larger-index runner-up overstates the gap (votes $(4, 6, 4)$ give $\mathrm{gap} = 1$, $r = 0$: one vote moved to class 0 yields $(5, 5, 4)$, which class 0 wins).

Proposition 1 (shard budget)**.** *If an adversary changes the votes of at most t realised shards, empties at most w and creates at most u new ones, the prediction is unchanged whenever $2t + u + w \leq \mathrm{gap}$. The bound is tight: some change of $r + 1$ votes flips it.*

Proposition 2 (row budgets)**.** *Assume A1–A4 and A6. An insertion, deletion or relabelling changes the multiset of one shard; a feature edit, with or without a relabelling, of at most two. Hence the prediction is unchanged under a insertions, deletions and relabellings plus b feature edits whenever $a + 2b \leq r$: under any $\rho \leq r$ operations of the first three kinds, and any $\rho \leq \lfloor r/2 \rfloor$ of all four.*

Proofs are in Appendix A. If a global stage lets one row influence g shards, every budget divides by g. The factor of two for feature edits comes from content addressing alone, since an edited row is re-hashed and may move.

Corollary 1 (identifier-keyed partition)**.** *If shards are assigned by $h(\mathrm{id})$ for row identifiers the adversary cannot reassign (a primary key, a signed log offset), inserted rows taking fresh ones, every operation changes one shard and the prediction is unchanged under any $\rho \leq r$ operations of any kind.*

Corollary 1 costs no measurable accuracy: an identifier hash spreads rows with distinct features as the content hash does (and spreads the duplicates content addressing co-locates), and over 20 tables it moves certified accuracy by a median +0.00 points, duplicate-heavy tables included (Appendix F). The halving is the price of a context without a trusted identifier, not a property of the learner. Neither result assumes h is secret: an adversary who computes h chooses *which* shard a row lands in, not how many; keying the hash removes that choice and an availability lever, and our attacks run unkeyed. The certificate has a structural ceiling, $r \leq \lfloor (m' - \mathbf{1}[c_1 > 0])/2 \rfloor$ for m' realised shards, so radius ρ needs $m' \geq 2\rho + \mathbf{1}[c_1 > 0]$ ($4\rho + \mathbf{1}[c_1 > 0]$ against feature edits): the shard count is the central design variable.

Predicting certified accuracy before partitioning. Fix a query q with true class c_q and let $p_q(n)$ be the probability that f predicts c_q from a uniformly random n-row subsample of S: the per-query learning curve, measurable with no partition.

Proposition 3 (learning-curve law)**.** *Assume that, conditional on q, the m shard votes are independent with $\Pr[v_s = c_q] = p_q(N/m)$. Then*

$$\Pr[\,q \textit{ correct and certified at radius } \rho\,] \;\geq\; \Pr\Big[\,\mathrm{Bin}\big(m,\, p_q(N/m)\big) \;\geq\; \Big\lceil \tfrac{m + 2\rho + \mathbf{1}[c_q > 0]}{2} \Big\rceil \Big], \tag{2}$$

with equality for $K = 2$ when all m shards are realised. Averaging over queries gives $\mathrm{CA}_{\mathrm{pred}}(\rho; m)$.

Independence is an approximation: shards are disjoint samples and their sizes are multinomial around N/m, so we measure the error (§8). The practical value is that $m^\star(\rho) = \arg\max_m \mathrm{CA}_{\mathrm{pred}}(\rho; m)$ is computable from the learning curve alone, without running the partition at any m.

4 Experimental protocol

Data and learners. 20 OpenML/UCI tables from the PMLB mirror (Olson et al., 2017; Romano et al., 2021; Vanschoren et al., 2014) (four synthetic: `ring`, `twonorm`, `waveform_21`, `banana`), with N from 800 to 10,000, d from 2 to 85, K from 2 to 10 and majority-class fractions from 0.11 to 0.95 (Appendix B). Seven in-context learners (TabPFN v2 (Hollmann et al., 2025), TabPFN v2.6, TabPFN v3, TabICL (Qu et al., 2025), TabICL v2, Mitra (Zhang et al., 2025) and Mitra v2) and five classical learners fitted per shard (logistic regression, random forest, XGBoost (Chen & Guestrin, 2016), an MLP, 15-NN) run under identical partition, certificate and aggregation code. We also ran TabPFN v3.5, the newest release (Table 1, Appendix P); it is reported separately and not pooled into the panel statistics. The benchmark, the main attack campaign and every timing ran on one NVIDIA A100 80 GB GPU. The baselines, on the same splits, are certified k-NN (Jia et al., 2022) over $k \in \{5, \ldots, 801\}$ with raw features or features standardised by the trusted query batch, and radius-NN over 15 query-scaled ball radii. Support-standardised k-NN is not certified (the adversary moves the statistics) and is excluded.

Grid and statistics. $m \in \{1, \ldots, 256\}$, radii to 128, seeds $\{0, 1, 2\}$: 6,372 completed cells, with 108 cells skipped where a table cannot supply 8 rows per shard. Comparisons are paired per table, one observation per table (seeds averaged inside it), reporting the median, a bootstrap 95% CI, win/tie/loss counts and a one-sided Wilcoxon signed-rank test (zero margins dropped), Holm-corrected (Holm, 1979) within each comparison. Under Table 2's protocol, leave-one-table-out singles out `ring`, where the margin is 37–45 points; without it the medians at radii 1–4 are $+0.56$ to $+0.89$ (12–13 wins of 19) and the means 1.9–2.4. Radii 0–24 are expressible by the grid and not still climbing at its boundary; we report no certified accuracy above radius 24 (Table 1's achieved median radii are unaffected).

Mechanical checks. We check A1 on every table and seed by appending and prepending rows: no row moves, each inserted row touches one shard and duplicates share a shard. Every learner fits its preprocessing inside the shard (A3), and every shard is evaluated in its own forward pass, so A5 holds by construction; TabPFN v2's fused multi-shard path is not vote-identical on GPU (about 0.1% of votes flip near ties), so we do not use it. Our tabular runs feed shard rows in arrival order and rely on row-permutation invariance, which we measure (Appendix E): Permuting rows within every shard changes 0.10–0.22% of TabPFN v2's shard votes (none of TabICL v2's or Mitra v2's; the other in-context learners were not tested), its radius on 1.4–11% of queries and 1 of 9,000 final predictions, so TabPFN v2's arrival-order certificates are not strictly sound. Sorting each shard by digest, then label, makes A3 hold by construction; re-certifying TabPFN v2 that way changes its certified accuracy by a median +0.00 points (at most 1.6; Appendix F).

5 What certification costs

Table 1 gives the price (Figure 3, Appendix G plots it). At $m = 16$, which buys a median radius of 7 rows, certification costs a median of 1.0 accuracy points for TabPFN v2 and 0.7–1.5 across the in-context panel (mean 1.9; the worst table, `credit_approval_germany` with 50 rows per shard, loses 6.4). At $m = 64$ (radius 31) the cost is 1.9–3.7 points, and at $m = 256$ (radius 106, about 2% of the median table's rows) 4.0–8.5. Learners differ in what they keep: at $m = 256$ TabICL v2 gives up 4.0 points, XGBoost 6.9 and 15-NN 14.6. The median table's shards hold about 20 rows there, so sharding is a sample-efficiency test. Voting buys back much of the per-shard loss (the plurality beats a single shard by 4.0 points at $m = 128$ and 5.4 at $m = 256$). Per-cell standard deviation over seeds is 0.8–1.3 points, so every comparison below is paired across tables. Compute is the other cost, and it depends on table size. Radius ρ needs $m \geq 2\rho + \mathbf{1}[c_1 > 0]$ forward passes, each over a context $1/m$ as long, and attention is superlinear in context length. On the benchmark tables per-call overhead wins: on an A100 80 GB the in-context panel's median undefended inference is 4.3 ms per query and the defended path 22.3, 74.2 and 282 ms at $m = 16$, 64 and 256, so a median radius of

Table 1: Cost of certification: median over 20 tables of seed-averaged values. Accuracy cost is undefended minus defended accuracy at the same shard count. The last rows give TabPFN v2's median radius against insertions, deletions and relabellings (r) and against any edit ($\lfloor r/2 \rfloor$); for $m \leq 32$ it is the same for every learner, at larger m it varies with shard agreement (28–31 at $m = 64$). †Reported separately, on one data split (Appendix P).

	accuracy cost (points) at $m =$						acc. (%) at $m =$		
Base learner	4	16	32	64	128	256	16	64	256
In-context									
TabPFN v2	0.4	1.0	1.8	3.0	4.3	6.9	93.3	91.9	87.0
TabPFN v2.6	0.4	1.2	2.5	3.1	4.5	6.7	92.9	92.3	88.4
TabPFN v3	0.3	1.1	2.1	3.0	4.0	7.1	93.1	92.5	87.4
TabICL	0.6	1.5	3.0	3.7	6.3	8.5	93.5	92.6	86.9
TabICL v2	0.3	0.9	1.6	2.5	3.2	4.0	93.7	93.0	89.6
Mitra	0.0	0.7	0.9	1.9	3.6	6.4	93.0	91.8	87.8
Mitra v2	0.3	0.9	2.2	3.2	3.9	7.9	94.2	92.2	88.4
TabPFN v3.5†	0.6	1.2	2.2	2.9	4.0	6.9	93.8	92.7	88.7
Classical									
LogReg	0.3	0.3	0.2	0.6	0.6	1.4	86.9	86.9	86.2
RF	1.0	2.1	2.2	3.0	4.1	8.3	92.5	90.7	85.6
XGBoost	0.7	1.8	3.0	4.4	6.1	6.9	93.0	89.9	83.8
MLP	0.1	1.0	1.6	2.5	3.7	6.2	93.1	90.5	85.6
15-NN	1.8	4.3	5.8	9.3	12.1	14.6	85.1	80.2	73.6
median radius r	2	7	15	31	59	106			
any-edit radius $\lfloor r/2 \rfloor$	1	3	7	15	29	53			

31 costs 17× (ratio of medians; 3× for Mitra and Mitra v2, 42× for TabPFN v3). At 20,000 rows attention wins (Appendix J): $m = 64$ costs TabPFN v2 2.0× its wall-clock and cuts its peak GPU memory 29×, TabICL's from 7.8 to 0.46 GB and Mitra v2's from 13.2 to 0.55 GB while running faster than undefended. The certificate is cheapest where the model is most expensive; certified k-NN is cheaper than any of them.

What content addressing costs. A1 forces a hash, which cannot balance shard sizes. Replacing it by a balanced random assignment (not certifiable, a control only) over TabPFN v2, TabICL, Mitra v2 and random forest, eight tables, $m \in \{8, \ldots, 128\}$ and three seeds changes accuracy by a median -0.02 points (IQR $[-0.21, +0.16]$; all 160 cells within one point; Appendix I). Subsample bagging with the probabilistic certificate of Jia et al. (2021) ($\max(100, m)$ draws) matches the partition's clean accuracy, but at matched m the deterministic partition leads it by a median 4.0 points of certified accuracy at every radius from 1 to 8, and by 2.1–5.3 points with each method at its best m (Appendix I). Against edits to existing rows' features the operative radius is $\lfloor r/2 \rfloor$ (Proposition 2), which costs a further 0.7, 0.9, 1.5 and 2.0 points of certified accuracy at radius 1, 2, 4 and 8 (median of per-table differences, best in-context configuration). That is the number to quote for a mutable store without identifiers; the insertion radius applies to an append-only or signed-feature store and to an identifier-keyed partition (Corollary 1).

6 Against certified k-NN

Certified k-NN (Jia et al., 2022) is retraining-free, per-query and deterministic, and certifies all four operations with one radius, since an inserted, deleted or edited row displaces at most one neighbour. The partition matches that radius without feature edits and halves it with them (Proposition 2), unless rows are keyed by identifier (Corollary 1). Table 2 therefore reports both threats, with radius-NN from the same paper. The comparison is held out: every benchmark cell stores each query's vote, radius and learning-curve correctness, so we split each table's queries into validation and test parts and make every choice on validation labels alone, without rerunning a model. The oracle version (each arm at its best configuration on the scored queries) and a 30/70 split with another split seed lead k-NN by 0.9–1.5 points at radii 1–4 and by under half a point at 8 (Appendix D).

Against insertions, deletions and relabellings, partitioned in-context prediction leads certified k-NN by $+0.67$, $+1.00$ and $+0.94$ points at radius 1, 2 and 4, on 13–14 of 20 tables (Holm $p \leq 0.043$). On

Table 2: Held-out paired margins in certified accuracy (points): median over 20 tables (seeds averaged within a table) and, in parentheses, tables won of 20. Each table's scored queries are split 40/60 per seed; on the validation part only the partition arms choose learner and shard count by the design rule of §8, k-NN its k and metric, radius-NN its ball radius; certified accuracy is read on the test part. ***i/d/r***: insertions, deletions and relabellings, where every method has radius r; ***any***: feature edits too, where k-NN and radius-NN keep r and the content-addressed partition has $\lfloor r/2 \rfloor$. One-sided Wilcoxon, Holm within each comparison (10 tests): $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$. ‡Needs partition radius 32, beyond the quotable grid; not read.

	edits	$\rho = 1$	$\rho = 2$	$\rho = 4$	$\rho = 8$	$\rho = 16$
In-context − k-NN	i/d/r	+0.67* (14)	+1.00* (14)	+0.94* (13)	+0.39 (12)	+0.00 (8)
	any	+0.06 (10)	+0.33 (11)	+0.06 (10)	-0.86 (6)	-2.78‡ (4)
In-context − radius-NN	i/d/r	+2.83*** (19)	+3.89*** (18)	+3.39*** (18)	+2.61** (17)	+2.11* (14)
	any	+2.28** (15)	+2.89** (17)	+2.22* (15)	+0.89 (13)	-0.33‡ (8)
In-context − classical	i/d/r	+0.61* (15)	+1.00** (14)	+1.36** (14)	+0.33 (14)	+0.78 (12)
	any	+1.00** (14)	+1.36** (14)	+0.33 (14)	+0.78 (12)	+0.06‡ (10)
Classical − k-NN	i/d/r	+0.28 (12)	+0.33 (12)	+0.17 (11)	-0.06 (9)	-0.33 (7)
	any	-0.22 (9)	-0.22 (9)	-0.94 (7)	-2.22 (5)	-4.00‡ (2)

five validation/test splits (Appendix D) the lead is significant at radius 2 on every split, at radius 1 on four and at radius 4 on one, and positive through radius 8 on all; we claim it to radius 2. Radius-NN trails by 2.6–3.9 points through radius 8.

The identical certificate over classical learners does not beat k-NN at any radius, while the in-context partition beats the classical one by $+0.61$, $+1.00$ and $+1.36$ points at radius 1, 2 and 4. The partition alone buys nothing over the k-NN certificate; the lead is the in-context learner's. Radius ρ forces $m \geq 2\rho + \mathbf{1}[c_1 > 0]$ shards of N/m rows, about 340 at $m = 16$ and 20 at $m = 256$ for the median table, and the in-context models are the ones that still work on a few dozen rows (Appendix G). Certified k-NN's radius is capped at $\lfloor k/2 \rfloor$ however large the table, while the partition's grows with the shards the table affords (Appendix K). Held out at 40,000 rows, the partition leads certified k-NN on `adult` and `shuttle` at radii 4 and 16 (by 0.3–3.0 points) and under arbitrary edits at radii 8 and 16 (by 1.0–5.0), where at 5,000 rows it trails on `adult` (Appendix L).

Under arbitrary edits the partition pays in radius, $\lfloor r/2 \rfloor$ against k-NN's r, and the lead is spent exactly: level at radii 1–4 ($+0.06$ to $+0.33$), behind at 8; the radius-16 cells need partition radius 32, beyond our quotable grid, and are not read. The learner's advantage is intact: under the same threat the in-context partition still beats the classical one by $+1.00$ and $+1.36$ at radii 1–2 and radius-NN to radius 4. What is lost is the accounting for a store without identifiers: with identifier-keyed shards (Corollary 1), TabPFN v2 alone leads certified k-NN under arbitrary edits by +0.83 and +1.50 points at radii 1 and 2 (16 and 15 of 20 tables), Holm-significant, and Mitra v2 alone to radius 4 (Appendix F).

Taken one learner at a time (oracle protocol, Table 6), every in-context learner has a positive median margin over k-NN at radius 1 ($+0.47$ to $+1.10$, 13–15 wins) and four survive a Holm correction over 48 tests; no classical learner does. No single model dominates: Mitra v2 is the best in-context learner on 8 tables at radius 1, TabICL v2 on 6, every other one on at least 2. The margin tends to grow with support size (Spearman $+0.31$ to $+0.40$ at radii 4–16, one observation per table), as the structural ceiling predicts, but no table property is significant after correction (Appendix M).

7 Adaptive attacks

Propositions 1–2 make the certificate sound whenever A1–A6 hold, so attacks test two things: whether the implementation meets the assumptions end to end, and how far the bound is from what an adversary needs. We ran two white-box attacks that know the model, the hash, the partition, the query and the current votes, against TabPFN v2, TabPFN v2.6, TabICL, Mitra v2 and 15-NN on six tables at $m \in \{1, 16, 64\}$: 2,250 queries, each attacked twice. *Grinding insertion* finds, per realised shard, the fewest inserted rows that flip its vote, each a copy of the query with the target label, perturbed in one coordinate until it hashes to that shard, by a doubling ladder and bisection that assume

a monotone per-shard response; it then buys the cheapest set of shards whose flip changes the plurality. *Label-only* relabels existing rows nearest the query, as in an append-only or crowd-labelled store. Both also run at $m = 1$, the source of the numbers in §1.

No insertion or label attack used r operations or fewer (Appendix H, Figure 4). Because the attack composes per-shard flips priced in isolation, that zero follows from its own accounting; the end-to-end test is to insert the composed rows into the full table and rerun the whole defended pipeline, which flipped 80 of 80 replays on three tables (Table 10). Eighty-three attacks did not flip within budget (73 insertion attacks at $m \geq 16$, capped at 64 rows per shard, and 10 label-only attacks at $m = 1$) and are censored, which can only raise the medians below. Minimum tightness is 1.00 in all ten (learner, m) configurations, at radii up to 32, and for the in-context learners attainment is typical: median insertion tightness is 1.00 at $m = 64$ for all four and at $m = 16$ for TabPFN v2.6, TabICL and Mitra v2. The mechanism is the one from §1, per shard: one copy of the query with the wrong label flips 92% of TabICL's shard votes, 70–75% of TabPFN v2.6's and Mitra v2's and 41% of TabPFN v2's, so the cheapest attack buys $r + 1$ shards at one row each. Label-only sits at 1.2–1.5 at $m = 64$ and 15-NN at 2.3–3.4, slack in the certificate for a learner whose shards need several copies to flip. A keyed hash would deny the choice of shard: simulated with the measured per-shard costs, a blind attacker needs a median 1.4–2.2× as many rows against the in-context learners.

Feature edits. The radius that halves under Proposition 2 needs its own attacks, each verified end to end. A features-only adversary rewrites existing rows without relabelling; its cheapest plans turn target-labelled rows into query copies and cost the insertion budget (median 1.00 of $r+1$ and 2.00 of $\lfloor r/2 \rfloor + 1$; 28 queries on TabPFN v2 and TabICL over `phoneme` ($m = 16, 64$)). An edit-and-relabel adversary, charged one operation by Proposition 2, can make an operation count twice: it takes a row nearest the query out of one victim shard and re-inserts it, relabelled, as a query copy in another. Removals flip a shard only after a median 18 rows (none within the cap for 33% of shards), copies after one, so it pairs operations on 15 of 49 successful attacks (52 queries over `magic` ($m = 16$), `phoneme` ($m = 16, 64$), `spambase` ($m = 16$)) and its median budget is 1.00 of $r + 1$. Yet on 8 of 49 it flipped the prediction with at most r operations, so the insertion radius does not survive edits, and on 1 of 49 it needed exactly $\lfloor r/2 \rfloor + 1$, attaining the halved radius. No attack of either kind succeeded within $\lfloor r/2 \rfloor$. TabICL v2, attacked on three tables, behaves like TabICL under insertion (90 of 90 flipped, median tightness 1.00; Table 9).

8 Predicting the certificate before you build it

Proposition 3 predicts the whole certified-accuracy curve from subsample accuracies, with no partition run. It consumes p_q, itself estimated from $R = 16$ subsample repeats, so even a perfect law shows error; we simulate that floor by resampling $\hat{p}_q \sim \mathrm{Bin}(R, p_q)/R$ and report observed error against it over 5,652 cells (Table 20, Appendix O). For in-context learners the MAE is 0.59–0.65 points against a floor of 0.31–0.34, leaving 0.20–0.23 points above the floor and a bias within ± 0.02; classical learners leave 0.30–0.40. Conditional independence fits a learner that reads its shard as a sample and fits no parameters to it better.

The part a deployer uses is the design rule, run once on trusted data (A1). In sample it has a median regret of zero for every in-context learner and picks the best shard count outright on 60–74% of cells; held out, with the curve measured on validation queries (Table 2), its median regret is 0.7–0.8 points at radii 1–4, the same as choosing learner and shard count by validation certified accuracy (0.6–0.8), which requires running every partition. The law's error concentrates at the structural ceiling, where it over-predicts: certification there needs every shard, the smallest included, to vote correctly, while the law gives each the success probability of an average one, which with $\log p_q(n)$ concave overstates the product (Jensen); fewer or smaller realised shards than nominal, and a curve measured on class-complete subsamples, push the same way. Inside the ceiling the median MAE falls from 0.72 to 0.59 points, and the error above the floor stays near 0.2 points until m exceeds 64.

9 The same certificate on an LLM

We replaced two components: the partitioner hashes an NFKC-normalised, whitespace-collapsed demonstration string without its label, and the backend scores each label by continuation log-probability; certificate, aggregation and law are unchanged. Six open models ran on an SST-2 subset (Socher et al., 2013) with 512 demonstrations, 200 queries and three seeds: Qwen3.5 at 2B, 4B and

9B, Llama-3.1-8B-Instruct, Gemma-4-E4B-It and Gemma-4-12B-It (Table 17). At $m = 32$ the first four are provably correct on 89.0, 88.0, 91.5 and 88.7% of queries against any 8 inserted, deleted or relabelled demonstrations of 512 (4 if their text may be edited), 1.0–5.7 points below undefended accuracy (a single prompt, capped at 4,096 tokens, so about 200 demonstrations survive at $m \leq 2$; shard prompts at $m \geq 4$ are never truncated). Their learning curves move at most 2.8 points from 4 to 256 demonstrations: with natural label words these models lean on priors (Min et al., 2022), the law says the shard count is free. Gemma-4-E4B-It is the case where it could: its curve rises from 67.5% at 4 demonstrations to 90.4% at 256, and the certificate charges for that dependence, 79.5% certified at radius 4 ($m = 16$) and 71.5% at radius 8 ($m = 32$) against 90.5% undefended; the law predicts 68.7% for that last cell from the curve alone. Gemma-4-12B-It scored near chance at every context length under our scoring path, a base-learner failure that we report and do not use.

Demonstrations that carry the task. To make every model depend on its demonstrations, we replaced the label words by `foo` and `bar`, the semantically unrelated setting of Wei et al. (2023): the only way to know which word means *positive* is to read the demonstrations the adversary may poison. For Qwen3.5-2B (512 demonstrations, 200 queries, two seeds, no truncation) the learning curve rises from 54.0% at 4 demonstrations to 90.4% at 256; undefended accuracy is 91.3%, the partition certifies 77.5% at radius 4 ($m = 16$) and 62.0% at radius 8 ($m = 32$), and the law predicts that last cell as 62.1% from the curve alone (Appendix N). A task learned from the demonstrations is exactly where poisoning bites and where certification is expensive: 16-row shards leave the model near the steep part of its curve, and the law says so before any partition is built.

A1 holds on text: inserting a demonstration moved no other and touched one shard. Hash-ordering demonstrations inside each prompt makes every prompt a function of its shard's set, so A3 holds by construction for autoregressive models; permuting the input list changed no prediction, which checks the implementation. Wall-clock at 32 shards is 2.4–8.8× that of one capped prompt.

10 Limitations

The radii are small in absolute terms (radius 8 is 0.16% of a 5,000-row table): the certificate addresses a few malicious contributors, not large-scale poisoning, and radii above 24 are truncated by our grid. The lead over certified k-NN is significant at radius 2 on every split (positive through 8) and only without feature edits; with them, and without row identifiers, the partition is level with k-NN at radii 1–4 and behind beyond. The k-NN baseline may standardise by the trusted query batch, which a per-request deployment would freeze. A certificate refers to one deterministic execution; our tabular backends rely on near-exact row-permutation invariance, and CPU and GPU runs can disagree near ties. The insertion attack covers five of seven in-context learners (label-only four, on six tables), the feature-edit attacks two learners on three tables. Finite Aggregation (Wang et al., 2022a) and Run-Off Election (Rezaei et al., 2023) would tighten the certificate unchanged; we did not evaluate them. Costs are wall-clock and peak memory; no FLOP counter supports these attention kernels. The LLM study is one sentiment task in two label regimes. We certify prediction integrity, not availability: an adversary can still degrade accuracy by unbalancing shard sizes, which a keyed hash mitigates.

11 Conclusion

Context handed to a model at inference time is part of its trusted computing base, and one inserted row can move a prediction. Hashing rows by their own content, voting over the shards and reading the vote gap tells the deployer, per query, how many inserted, deleted or relabelled rows would change the answer, at a median cost of one accuracy point for a radius of seven, and the learning curve says what it will cost before any partition is built. The rule the evidence supports: for an append-only, signed or identifier-keyed store, partition the in-context model and gain about a point over certified k-NN at radii 1–2, a radius that grows with the table size the deployment is designed for and, at 20,000 rows, a smaller GPU; for a mutable store without identifiers, certified k-NN gives the same certified accuracy for a fraction of the compute. Swap the partitioner and the backend and the same certificate covers an LLM reading demonstrations.

ETHICS STATEMENT

This work is defensive: it gives deployers of in-context predictors a way to bound how much poisoned context can change a prediction. The attacks we describe are simple and white-box (copy the query, attach a target label, insert it), are close to what the poisoning literature already describes, and serve only to test the certificate; we judge that describing them adds little capability for an attacker next to the value of a tested defence. The certificate holds only under the stated assumptions and for a bounded number of changed rows, and it certifies that a prediction is invariant, not that it is correct or fair; overstating it could give a false sense of security, so we state its limits in §10. All tabular data are public benchmark tables, some with de-identified demographic attributes, used only as benchmarks, and the language experiments use the public SST-2 corpus. No human subjects were involved and no personal data were collected.

REPRODUCIBILITY STATEMENT

We provide anonymised code, submitted as supplementary material, that implements the partition, the certificate, the interfaces to every model, the attacks, the learning-curve law and every experiment. All tables come from the public PMLB mirror of OpenML and UCI data and the language experiments use public SST-2; every model is a publicly released checkpoint. Splits are deterministic functions of the seed, and the hash is BLAKE2b with a fixed personalisation string and no key, so every shard assignment can be recomputed exactly. Grids, seeds, baselines and statistical tests are given in §4, and proofs of every statement in Appendix A. Because each query's votes, radii and learning-curve correctness are recorded, the held-out analysis of §6 can be repeated without rerunning any model. The benchmark, the main attack campaign and all timing measurements ran on one NVIDIA A100 80 GB GPU.

AI USE STATEMENT

We used generative AI tools for language editing, LaTeX and figure preparation, auditing and writing analysis code, and discussing hypotheses and the interpretation of results. The authors ran the experiments, checked every reported number against the saved records, and take full responsibility for the claims, code, and submission.

REFERENCES

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *International Conference on Machine Learning (ICML)*, 2012.

Ruoxin Chen, Zenan Li, Jie Li, Chentao Wu, and Junchi Yan. On collective robustness of bagging against data poisoning. 2022. URL `https://arxiv.org/abs/2205.13176`.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.

Mohamed Djilani, Thibault Simonetto, Karim Tit, Florian Tambon, Paul Récamier, Salah Ghamizi, Maxime Cordy, and Mike Papadakis. On the robustness of tabular foundation models: Test-time attacks and in-context defenses. *arXiv preprint arXiv:2506.02978*, 2025.

Zayd Hammoudeh and Daniel Lowd. Provable robustness against a union of ℓ_0 adversarial attacks. In *AAAI Conference on Artificial Intelligence*, 2024.

Pengfei He, Han Xu, Yue Xing, Hui Liu, Makoto Yamada, and Jiliang Tang. Data poisoning for in-context learning. In *Findings of the Association for Computational Linguistics: NAACL*, 2025.

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

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

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. In *AAAI Conference on Artificial Intelligence*, 2021.

Jinyuan Jia, Yupei Liu, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of nearest neighbors against data poisoning and backdoor attacks. In *AAAI Conference on Artificial Intelligence*, 2022.

Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *International Conference on Learning Representations (ICLR)*, 2021.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

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

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning (ICML)*, 2025.

Zhiyao Ren, Siyuan Liang, Aishan Liu, and Dacheng Tao. ICLShield: Exploring and mitigating in-context learning backdoor attacks. In *International Conference on Machine Learning (ICML)*, 2025.

Keivan Rezaei, Kiarash Banihashem, Atoosa Chegini, and Soheil Feizi. Run-off election: Improved provable defense against data poisoning attacks. In *International Conference on Machine Learning (ICML)*, 2023.

Joseph D. Romano, Trang T. Le, William La Cava, John T. Gregg, Daniel J. Goldberg, Natasha L. Ray, Praneel Chakraborty, Daniel Himmelstein, Weixuan Fu, and Jason H. Moore. Pmlb v1.0: An open source dataset collection for benchmarking machine learning methods. 2021. URL https://arxiv.org/abs/2012.00058.

Yan Scholten, Jan Schuchardt, Aleksandar Bojchevski, and Stephan Günnemann. Hierarchical randomized smoothing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Wenxiao Wang, Alexander Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *International Conference on Machine Learning (ICML)*, 2022a.

Wenxiao Wang, Alexander Levine, and Soheil Feizi. Lethal dose conjecture on data poisoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Yanting Wang, Wei Zou, and Jinyuan Jia. FCert: Certifiably robust few-shot classification in the era of foundation models, 2024. URL https://arxiv.org/abs/2404.08631.

Maurice Weber, Xiaojun Xu, Bojan Karlaš, Ce Zhang, and Bo Li. RAB: Provable robustness against backdoor attacks. In *IEEE Symposium on Security and Privacy (S&P)*, 2023.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently, 2023. URL https://arxiv.org/abs/2303.03846.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. Certifiably robust RAG against retrieval corruption, 2026. URL https://arxiv.org/abs/2405.15556.

Xiyuan Zhang, Danielle C. Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W. Mahoney, Cuixiong Hu, Huzefa Rangwala, George Karypis, and Bernie Wang. Mitra: Mixed synthetic priors for enhancing tabular foundation models. 2025. URL https://arxiv.org/abs/2510.21204.

Shuai Zhao, Meihuizi Jia, Anh Tuan Luu, Fengjun Pan, and Jinming Wen. Universal vulnerabilities in large language models: Backdoor attacks for in-context learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

A PROOFS

Proof of Proposition 1. A changed vote lowers n_{c_1} by at most one and raises at most one other count by one; an emptied shard lowers one count by one; a new shard raises one count by one. Hence $n'_{c_1} \geq n_{c_1} - t - w$ and $n'_c \leq n_c + t + u$ for every $c \neq c_1$, so $n'_{c_1} - n'_c - \mathbf{1}[c < c_1] \geq \text{gap} - (2t + u + w) \geq 0$ by equation 1, which is exactly the condition for c_1 to remain the plurality against every rival c under the smaller-index tie rule. For tightness, let $c^\star$ attain the minimum in equation 1 and move $r + 1 \leq n_{c_1}$ votes from c_1 to $c^\star$: the difference becomes $\text{gap} + \mathbf{1}[c^\star < c_1] - 2(r+1) < \mathbf{1}[c^\star < c_1]$. □

Proof of Proposition 2. By A1 and A4 a row's shard is a fixed function of its own canonical features, so inserting or deleting a row alters only shard $h(x)$, a relabelling alters only $h(x)$ because the label is not hashed, and an edit $x \to x'$ alters only $h(x)$ and $h(x')$. By A3 and A6 every unaltered shard casts the same vote. Each altered shard is a changed vote (cost 2 in Proposition 1), an emptied shard or a new one (cost 1), so a operations of the first kinds and b feature edits cost at most $2a + 4b \leq 2r \leq \text{gap}$. □

Proof of Corollary 1. An edit leaves the identifier, hence the shard, unchanged; an insertion or deletion touches the shard of its own identifier. Each operation therefore alters one shard, at cost at most 2, and Proposition 1 applies with $2\rho \leq \text{gap}$. □

Proof of Proposition 3. Let V be the number of shards voting c_q. Every wrong vote goes to some $c \neq c_q$, so $n_c \leq m - V$, and $\mathbf{1}[c < c_q] \leq \mathbf{1}[c_q > 0]$; hence $n_{c_q} - n_c - \mathbf{1}[c < c_q] \geq 2V - m - \mathbf{1}[c_q > 0]$ for every $c \neq c_q$. On the event on the right of equation 2 this is at least 2ρ, so c_q is the prediction with $\text{gap} \geq 2\rho$. For $K = 2$ the wrong class receives exactly $m - V$ votes and $\mathbf{1}[c < c_q] = \mathbf{1}[c_q > 0]$, so the events coincide. The inclusion is distribution-free; independence is used only to evaluate the right-hand side as a Binomial tail. □

B BENCHMARK TABLES

Table 3 lists the panel. Support and query rows are split by seed, support is capped at 10,000 rows and queries at 500, and every table is used as the source ships it: categorical columns stay as integer codes, with no imputation, scaling or deduplication.

Table 3: The 20 benchmark tables. N is the support size after the split, Q the number of held-out queries, "cat." the number of categorical columns, "maj." the majority-class fraction.

Table	N_{tot}	d	K	cat.	N	Q	maj.
shuttle	58000	9	7	0	10000	500	0.78
adult	48842	14	2	5	10000	500	0.76
magic	19020	10	2	0	10000	500	0.64
nursery	12958	8	4	8	10000	500	0.33
pendigits	10992	16	10	0	10000	500	0.11
coil2000	9822	85	2	84	9322	500	0.94
mushroom	8124	22	2	21	7624	500	0.51
ring	7400	20	2	0	6900	500	0.50
twonorm	7400	20	2	0	6900	500	0.50
satimage	6435	36	6	0	5935	500	0.24
page_blocks	5473	10	5	0	4973	500	0.90
phoneme	5404	5	2	0	4904	500	0.71
banana	5300	2	2	0	4800	500	0.55
waveform_21	5000	21	3	0	4500	500	0.34
churn	5000	20	2	4	4500	500	0.86
spambase	4601	57	2	0	4101	500	0.61
kr_vs_kp	3196	36	2	36	2696	500	0.52
hypothyroid	3163	25	2	18	2663	500	0.95
titanic	2207	8	2	5	1766	441	0.68
credit_approval_germany	1000	20	2	17	800	200	0.70

C IMPLEMENTATION

Canonical encoding. Rows are encoded by casting to float32 and taking the raw little-endian bytes, with -0.0 mapped to 0.0 and NaN payloads normalised. Float32 rather than absolute decimal rounding: rounding to a fixed number of decimals makes the assignment sensitive to representation differences that survive a round trip through a different pipeline, which breaks reproducibility without buying anything for soundness.

Assumption checks. For every table and seed we append and prepend held-out rows to the support table and recompute the assignment of every original row. Across all $20 \times 3 = 60$ table–seed pairs no row moved on append or on prepend, each inserted row touched exactly one shard, and identical rows always shared a shard. Every learner fits its preprocessing inside the shard (A3), and every shard is evaluated in its own forward pass, so A5 holds by construction.

Certificate verification. For vote vectors small enough to enumerate, the implementation exhaustively checks that no reassignment of r votes changes the plurality and that some reassignment of $r + 1$ votes does. This confirms the shard-level bound is tight as arithmetic, independently of the attack experiments.

Grid truncation. For each radius we compute the fraction of (base learner, table) cells whose best shard count sits at the top of the grid and whose certified accuracy is still rising there. A radius is quotable only when both fractions are below one half and both arms can structurally express it. Cells that certify nothing at a radius are excluded from the fractions, because an arbitrary argmax over a set of zeros otherwise makes a truncated radius look safe. Under this rule radii 0–24 are quotable at $m \leq 256$, $k \leq 801$; radii 32 and above are not.

D SELECTION CONTROLS: ORACLE COMPARISON AND SINGLE LEARNERS

Table 4 lets each arm choose on the queries it is scored on, which is symmetric but optimistic for both. Its margins are a little larger than the held-out ones and tell the same story: the in-context partition leads k-NN by about one point to radius 4 against insertions, deletions and relabellings, the classical partition does not lead at any radius after correction, and under arbitrary edits the two

Table 4: Oracle version of Table 2: each arm at its best configuration per table on the scored queries (learner and shard count for the partition arms; k and certified metric for k-NN), paired over 20 tables with seeds averaged within a cell. Stars: one-sided Wilcoxon, Holm-corrected over all 25 tests: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$. ‡Needs partition radius 32, beyond the quotable grid.

	$\rho = 1$	$\rho = 2$	$\rho = 4$	$\rho = 8$	$\rho = 16$
Threat: insertions, deletions, relabellings (partition radius r)					
In-context partition − certified k-NN					
median margin	+1.13**	+1.10*	+1.00*	+0.47	+0.17
95% CI	[+0.5, +4.2]	[+0.2, +4.3]	[+0.0, +2.1]	[+0.0, +1.3]	[+0.0, +1.9]
win/tie/loss	16/2/2	14/5/1	13/4/3	11/4/5	12/3/5
Classical partition − certified k-NN					
median margin	+0.37	+0.38	+0.13	+0.00	-0.07
win/tie/loss	14/1/5	11/2/7	10/2/8	8/3/9	7/3/10
In-context partition − classical partition					
median margin	+0.80*	+1.03**	+0.90*	+0.27	+0.37
win/tie/loss	16/1/3	16/2/2	16/2/2	14/3/3	12/3/5
Threat: any row edits incl. features (partition radius $\lfloor r/2 \rfloor$)					
In-context partition − certified k-NN					
median margin	+0.50	+0.17	-0.30	-0.53	-1.93‡
win/tie/loss	12/2/6	10/3/7	7/2/11	5/3/12	4/4/12
Classical partition − certified k-NN					
median margin	-0.10	+0.00	-0.23	-1.10	-2.91‡
win/tie/loss	9/1/10	9/2/9	8/2/10	4/3/13	3/3/14

are level at radius 1–2 and k-NN is ahead beyond. Averaging the raw and query-scaled k-NN arms before choosing k would weaken the baseline; here k-NN takes the better certified metric per table, as the partition arm takes the better learner.

Table 5 repeats the held-out comparison of Table 2 on four further random 40/60 splits of the scored queries. The lead over certified k-NN against insertions, deletions and relabellings is positive at every radius on every split; it is Holm-significant at radius 2 on all five splits, at radius 1 on four and at radius 4 on one. Under arbitrary edits it is never significant.

Table 5: In-context partition minus certified k-NN, held out, on five validation/test splits (median over 20 tables; Holm within each split as in Table 2).

	ins./del./rel.				any edit	
split seed	$\rho = 1$	$\rho = 2$	$\rho = 4$	$\rho = 8$	$\rho = 1$	$\rho = 2$
12345 (Table 2)	+0.67*	+1.00*	+0.94*	+0.39	+0.06	+0.33
1	+1.56**	+1.28**	+0.89	+0.22	+0.61	+0.78
2	+0.89*	+1.17*	+0.78	+0.17	+0.50	+0.44
3	+0.78	+0.72*	+0.67	+0.33	+0.06	+0.44
4	+1.28**	+0.61*	+0.39	+0.28	+0.06	+0.33

Figure 2 draws the held-out comparisons of Table 2 with their confidence intervals. Against insertions, deletions and relabellings the in-context partition is at or above certified k-NN and above the classical partition at every radius, and above radius-NN by two to four points. Under arbitrary edits it is level with certified k-NN, and its leads over the classical partition (radii 1–2) and radius-NN (radii 1–4) are significant.

The in-context arm takes the best of seven learners per table and the classical arm the best of five. Table 6 removes that selection. Every in-context learner has a positive median margin at radius 1 ($+0.47$ to $+1.10$, 13–15 wins) and four (TabPFN v2, TabPFN v3, TabICL v2, Mitra v2) survive the correction; no classical learner does, and 15-NN wins on no table. At small radii the ordering comes from the model class, not from selecting across it.

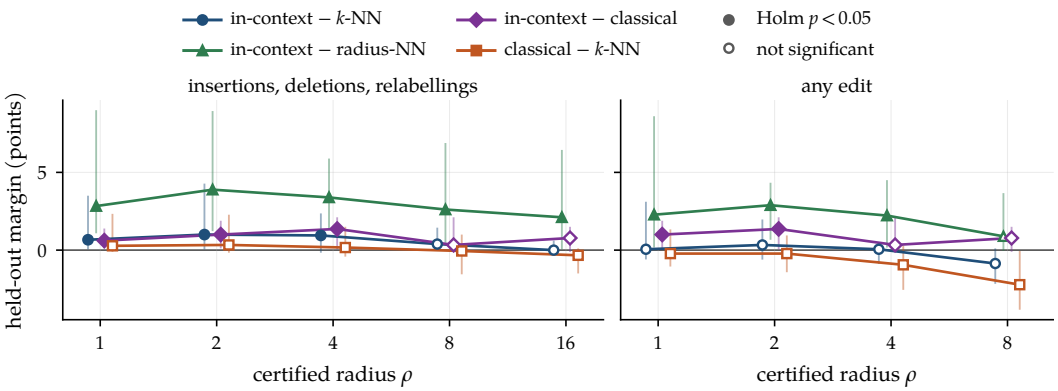


Figure 2: Held-out margins of Table 2: median over 20 tables with a bootstrap 95% interval, against insertions, deletions and relabellings (left) and any edit (right). Filled markers are Holm-significant ($p < 0.05$) within their comparison, hollow ones are not. Any edit at $\rho = 16$ needs partition radius 32, beyond the quotable grid, and is not drawn.

Table 6: Each base learner on its own against certified k-NN (insertions, deletions and relabellings): the learner's best shard count per table against the best k and metric, paired over 20 tables with seeds averaged within a cell. Stars: one-sided Wilcoxon, Holm-corrected over all 48 tests.

	$\rho = 1$		$\rho = 2$		$\rho = 4$		$\rho = 8$	
Base learner	margin	W/T/L	margin	W/T/L	margin	W/T/L	margin	W/T/L
In-context								
TabPFN v2	+0.83*	14/2/4	+0.57*	14/4/2	+0.23	11/3/6	-0.13	6/2/12
TabPFN v2.6	+0.87	14/1/5	+0.60	13/2/5	+0.03	10/3/7	+0.00	8/3/9
TabPFN v3	+0.97*	15/2/3	+0.90*	13/4/3	+0.67	11/2/7	+0.20	11/3/6
TabICL	+0.80	14/1/5	+0.77	12/3/5	+0.33	10/3/7	+0.00	7/4/9
TabICL v2	+1.10*	15/1/4	+0.77	14/1/5	+0.60	12/2/6	+0.27	10/2/8
Mitra	+0.47	13/1/6	+0.53	11/3/6	+0.07	10/2/8	+0.00	8/3/9
Mitra v2	+1.03*	14/1/5	+0.97	14/2/4	+0.47	13/3/4	+0.10	10/3/7
Classical								
LogReg	-0.77	6/1/13	-0.73	5/2/13	-0.57	6/1/13	-0.97	6/2/12
RF	+0.13	12/1/7	-0.18	8/1/11	-0.18	8/1/11	-0.33	6/2/12
XGBoost	+0.07	10/1/9	+0.00	9/2/9	-0.43	7/2/11	-0.28	6/2/12
MLP	+0.15	10/1/9	+0.08	10/2/8	-0.37	6/1/13	-1.07	4/2/14
15-NN	-2.02	0/1/19	-2.86	0/2/18	-4.60	0/2/18	-6.04	0/3/17

E Row order and determinism

Row order is not a counted operation, so an adversary may permute the support table for free, and A3 requires each shard's vote to depend on its rows as a multiset. For TabPFN v2, TabICL v2 and Mitra v2 on `phoneme`, `adult` and `satimage` at $m \in \{16, 64\}$ (500 queries each), we evaluated every realised shard four times: in arrival order, again in arrival order, with its rows permuted uniformly at random, and with its rows sorted by content digest. Table 7 reports the fraction of shard votes that differ from the first run and the number of final predictions that change.

F Identifier-keyed partition and canonical re-certification

For TabPFN v2 and Mitra v2 (seed 0, $m \in \{4, \ldots, 64\}$; TabPFN v2 on 20 tables, Mitra v2 on 20 tables) we reran the partition twice with every shard's rows sorted by digest and then label, so that A3 holds by construction: once with the content hash and once with the identifier hash of Corollary 1, the identifier being a row's position in the split, fixed before any attack. Against the

Table 7: Row-order and determinism check: shard votes changed (of all shard-query votes) and final predictions changed, relative to a first run in arrival order.

			repeat		rows permuted		rows hash-sorted	
Learner	table	m	votes	preds	votes	preds	votes	preds
TabPFN v2	adult	16	0.00%	0	0.10%	0	0.14%	1
TabPFN v2	adult	64	0.00%	0	0.18%	0	0.18%	0
TabPFN v2	phoneme	16	0.00%	0	0.19%	0	0.18%	0
TabPFN v2	phoneme	64	0.00%	0	0.22%	1	0.27%	0
TabPFN v2	satimage	16	0.00%	0	0.10%	0	0.07%	0
TabPFN v2	satimage	64	0.00%	0	0.18%	0	0.18%	0
Mitra v2	adult	16	0.00%	0	0.00%	0	0.00%	0
Mitra v2	adult	64	0.00%	0	0.00%	0	0.00%	0
Mitra v2	phoneme	16	0.00%	0	0.00%	0	0.00%	0
Mitra v2	phoneme	64	0.00%	0	0.00%	0	0.00%	0
Mitra v2	satimage	16	0.00%	0	0.00%	0	0.00%	0
Mitra v2	satimage	64	0.00%	0	0.00%	0	0.00%	0
TabICL v2	adult	16	0.00%	0	0.00%	0	0.00%	0
TabICL v2	adult	64	0.00%	0	0.00%	0	0.00%	0
TabICL v2	phoneme	16	0.00%	0	0.00%	0	0.00%	0
TabICL v2	phoneme	64	0.00%	0	0.00%	0	0.00%	0
TabICL v2	satimage	16	0.00%	0	0.00%	0	0.00%	0
TabICL v2	satimage	64	0.00%	0	0.00%	0	0.00%	0

arrival-order cells of the main benchmark, the canonical content partition changes TabPFN v2's certified accuracy by a median +0.00 points (largest change 1.6 points; this bounds the order effect and any hardware difference between the two runs together). Hashing the identifier instead of the content changes it by a median +0.00 points; on `titanic` and `coil2000`, whose duplicate rows content addressing co-locates and an identifier hash spreads, the medians are +0.7 and +0.0 points. Because every operation touches one shard under the identifier hash, its radius holds against any edit; Table 8 compares both partitions with certified k-NN under arbitrary edits on the held-out protocol of Table 2. With identifier-keyed shards either learner alone leads certified k-NN under arbitrary edits at radii 1 and 2 (Holm-significant; Mitra v2 also at radius 4), where its content-addressed partition is level: the halving of §3 is the only thing that stands between a mutable store and the insertion-threat lead of Table 2.

Table 8: Held-out margin in certified accuracy over certified k-NN under *any* edit (points; median over tables, tables won in parentheses) for one learner's content-addressed and identifier-keyed partitions, both with shards sorted by digest and label, $m \in \{4, \ldots, 64\}$, seed 0. Holm within each learner (8 tests).

learner	partition	$\rho = 1$	$\rho = 2$	$\rho = 4$	$\rho = 8$
TabPFN v2	content	+0.33 (11)	+0.17 (10)	-0.75 (5)	-2.17 (5)
	keyed	+0.83** (16)	+1.50** (15)	+0.52 (12)	+0.00 (8)
Mitra v2	content	+0.50 (12)	+0.50 (12)	+0.00 (10)	-2.00 (4)
	keyed	+1.33** (16)	+0.75* (13)	+1.50* (13)	+0.17 (11)

G COST FRONTIER

Figure 3 plots the cost of Table 1: accuracy lost to partitioning against shard count, accuracy against achieved median radius with certified k-NN on the same axes, and rows per shard against m.

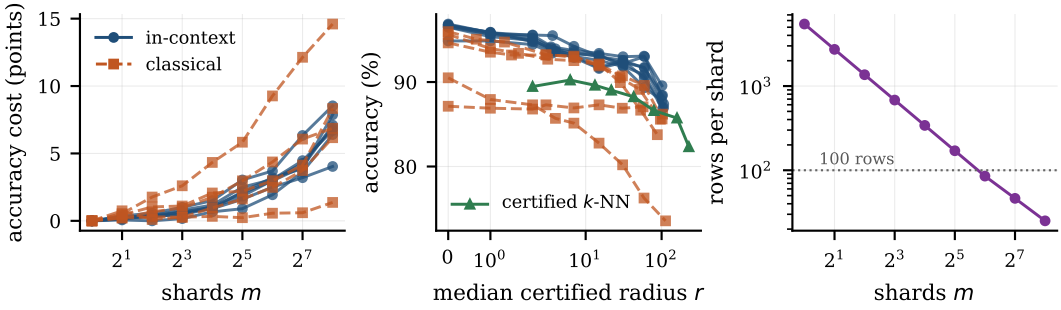


Figure 3: **Left:** accuracy cost grows smoothly with the shard count, and at $m = 256$ runs from 4.0 points (TabICL v2) to 14.6 (15-NN), with logistic regression lower only because it starts 9 points below the rest. **Middle:** accuracy against achieved certified radius, with certified k-NN (green) on the same axes. **Right:** the shard count sets rows per shard, the quantity the base learner actually sees.

Table 9: Adaptive attacks against five base learners on six tables, pooled over tables. n_q counts queries the attack flipped within its budget; budget is the median number of operations it needed and tightness the median budget$/(r+1)$, so 1.00 means the certificate is exactly attained. No successful insertion or label-only attack used at most r operations. [§]Attacked separately on `phoneme`, `magic` and `satimage` (15 queries each, insertion only).

		insertion				label flip			
Base learner	m	n_q	med. r	budget	tight.	n_q	med. r	budget	tight.
TabPFN v2	16	164	7	14	1.71	180	7	29	3.35
TabPFN v2	64	180	28	32	1.00	180	28	42	1.52
TabPFN v2.6	16	160	7	9	1.00	180	7	24	3.12
TabPFN v2.6	64	179	28	30	1.00	180	28	39	1.37
TabICL	16	158	7	8	1.00	180	7	21	2.53
TabICL	64	180	28	29	1.00	180	28	33	1.22
Mitra v2	16	166	7	9	1.00	180	7	26	3.06
Mitra v2	64	180	27	28	1.00	180	27	37	1.36
15-NN	16	180	7	36	4.50	180	7	26	3.40
15-NN	64	180	26	92	3.41	180	26	60	2.26
TabICL v2[§]	16	45	7	8	1.00	–	–	–	–
TabICL v2[§]	64	45	26	27	1.00	–	–	–	–

H ATTACK DETAIL

Table 9 gives the campaign by learner, shard count and operation, and Figure 4 plots every attacked query. The 15-NN rows are the useful contrast: its median insertion attack at $m = 64$ needs 3.4 times $r + 1$ rows, where every in-context learner is flipped at exactly $r + 1$.

Feature edits and end-to-end replay. Table 10 reports the feature-edit adversary of §7 (top: features of existing rows only; bottom: features and label together) and, on the same queries, the grinding insertion attack replayed end to end: its composed rows are inserted into the full table, the table is re-partitioned from scratch and every shard re-votes. Support tables hold 4,000 rows as in Table 9; per-shard budgets are capped at 64 rows, and unflipped queries are censored.

Figure 5 plots every flipped query of Table 10. Feature edits alone land on or above the insertion budget $r + 1$; edits paired with relabels drop below it on 8 of 49 and reach the halved bound once. No point enters the region that would falsify the any-edit certificate.

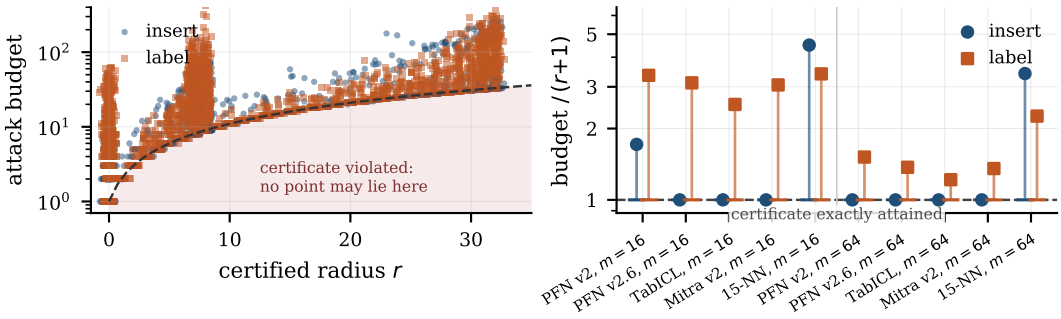


Figure 4: **Left:** every attacked query, budget against certified radius. The shaded region is where a point would falsify the certificate; it is empty. **Right:** median (marker) and minimum (bar) tightness per configuration. The minimum is 1.00 everywhere, so in every configuration some query is flipped at exactly $r + 1$ operations.

Table 10: Feature-edit attacks against the mixed radius $\lfloor r/2 \rfloor$ and end-to-end replay of the insertion attack on the same queries. Top: features only; bottom: features and label in one operation. *viol.*: successful attacks using at most $\lfloor r/2 \rfloor$ operations (a certificate violation).

			feature-edit attack					insertion replay	
Learner	table	m	n_q	flipped	med. $\lfloor r/2 \rfloor$	med. budget	viol.	replayed	flipped
TabPFN v2	`phoneme`	16	15	12	3	8	0	15	15
TabPFN v2	`phoneme`	64	3	3	16	59	0	3	3
TabICL	`phoneme`	16	10	10	4	9	0	10	10
edit-and-relabel attack									
TabPFN v2	`magic`	16	10	9	3	12	0	10	10
TabPFN v2	`phoneme`	16	10	8	4	11	0	10	10
TabPFN v2	`phoneme`	64	3	3	16	52	0	3	3
TabPFN v2	`spambase`	16	10	10	4	14	0	10	10
TabICL	`phoneme`	16	10	10	4	9	0	10	10
TabICL	`spambase`	16	9	9	4	9	0	9	9

I WHAT CONTENT ADDRESSING COSTS

Table 11 gives the control per learner and shard count. A content-addressed hash cannot balance shard sizes, so the partition it produces is more uneven than a random one of the same granularity, and unevenness costs accuracy. The control isolates that cost: both arms use the same base learner, the same support and query split, and the same nominal shard count, so the only difference is how rows are assigned. Across TabPFN v2, TabICL, Mitra v2 and random forest, eight tables and five shard counts, with three seeds averaged inside each of the 160 cells, the median difference is -0.02 points, the interquartile range is $[-0.21, +0.16]$, every cell falls within one point either way, and the largest single difference is 0.93. Whatever limits the method, it is not the shard-size imbalance a hash induces, and that holds for the in-context learners the paper is about as well as for the classical control.

Subsample bagging draws overlapping subsamples rather than disjoint shards and certifies with the probabilistic guarantee of Jia et al. (2021). In clean accuracy it is level with the content-addressed partition (median difference 0.00, within ± 0.5 at every shard count). In certified accuracy it is not: Table 12 shows the deterministic partition ahead by a median 4.0 points at every radius from 1 to 8 over the same 160 cells at matched m (2.1, 2.4, 3.6 and 5.3 points at radius 1, 2, 4 and 8 with each method at its own best m; bagging uses $\max(100, m)$ draws at $\alpha = 10^{-3}$), while the balanced random control stays within 0.1 points of content addressing at every radius. The deterministic guarantee therefore costs nothing against either control and buys certified accuracy against bagging.

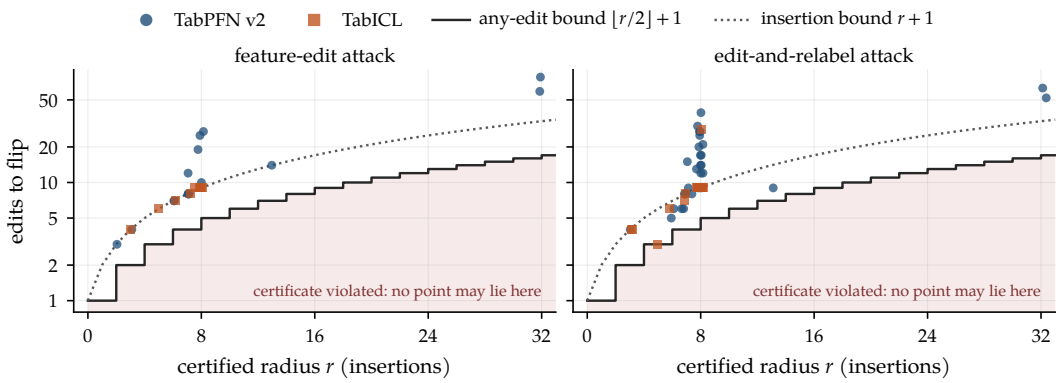


Figure 5: Edits needed to flip each flipped query against its insertion radius r, for the feature-edit (left) and edit-and-relabel (right) attacks on TabPFN v2 and TabICL. Solid step: the any-edit guarantee $\lfloor r/2 \rfloor + 1$; dotted: the insertion guarantee $r + 1$. The shaded region, where a point would falsify the certificate, is empty.

Table 11: Content-addressed partition minus a balanced random partition of the same granularity and minus subsample bagging, in accuracy points, median over eight tables with three seeds averaged inside each cell. The random arm is not certifiable in this setting (it violates A1) and is included only as a control for the shard-size imbalance a hash induces; bagging certifies with a probabilistic guarantee.

		at $m =$				
Base learner	cells	8	16	32	64	128
content − random (points)						
TabPFN v2	40	-0.01	-0.08	+0.12	+0.07	+0.24
TabICL	40	+0.02	+0.00	-0.14	-0.10	-0.11
Mitra v2	40	-0.07	-0.02	+0.03	-0.02	-0.07
RF	40	+0.07	-0.04	-0.09	-0.04	+0.02
all	160	-0.01	-0.02	-0.07	+0.00	-0.07
content − bagging (points)						
TabPFN v2	40	-0.23	-0.00	+0.00	+0.43	+0.37
TabICL	40	+0.03	-0.13	-0.07	+0.17	+0.17
Mitra v2	40	-0.17	-0.13	-0.07	-0.07	+0.07
RF	40	-0.03	-0.03	-0.07	-0.07	+0.33
all	160	-0.13	-0.07	+0.00	+0.10	+0.23

J SYSTEMS COST

Table 13 gives wall-clock and peak memory at two table sizes. Memory is the clearer of the two effects. An in-context model attends over its whole context, so peak memory grows with N; the partition caps the context at N/m rows, so past a modest m the peak is flat in N. At 20,000 rows at $m = 64$ the reduction is 12× for TabPFN v2.6 (1.1 GB to 95 MB), 17× for TabICL (7.8 GB to 0.46 GB), 24× for Mitra v2 (13.2 GB to 0.55 GB) and 29× for TabPFN v2 (2.3 GB to 79 MB). Time depends on scale. At 1,000 rows the defended path is slower by roughly the shard count, because each shard is small enough that fixed per-call overhead dominates. At 20,000 rows the superlinear cost of attention over the full context takes over: $m = 16$ is faster than the undefended TabPFN v2 (10.2 s against 15.6 s) and $m = 64$ costs 2.0×, and Mitra v2 is faster defended than undefended at every m up to 64 (10.1 s at $m = 64$ against 12.9 s). Relative to the model it protects, the certificate is cheapest on the tables where that model is most expensive.

Table 12: Content-addressed partition minus the balanced random partition and minus subsample bagging in *certified* accuracy (points) at radius ρ: median and interquartile range over the 160 (learner, table, m) cells of Table 11, seeds averaged within a cell. Bagging's radius is certified with confidence $1 - \alpha$ over its subsample draw.

	content − random		content − bagging	
ρ	median	IQR	median	IQR
0 (clean)	-0.02	[-0.2, +0.2]	+0.00	[-0.2, +0.2]
1	-0.02	[-0.2, +0.2]	+4.07	[+2.3, +6.0]
2	-0.07	[-0.3, +0.1]	+4.00	[+2.5, +6.0]
4	-0.02	[-0.4, +0.1]	+4.00	[+2.1, +6.2]
8	+0.00	[-0.3, +0.0]	+4.00	[+0.0, +6.5]

Table 13: Wall-clock (seconds, 64 queries) and peak GPU memory (MB) for the defended path on synthetic 20-column tables, median over three repeats on an A100 80 GB. $m = 1$ is the undefended model. The FLOP counter does not support these attention kernels, so operation counts are not reported.

		$m = 1$		$m = 16$		$m = 64$		$m = 256$	
Base learner	N	s	MB	s	MB	s	MB	s	MB
TabPFN v2	1,000	1.1	151	8.3	68	29.4	68	86.4	68
TabPFN v2	20,000	15.6	2262	10.2	186	31.3	79	120.1	68
TabPFN v2.6	1,000	2.0	110	13.5	95	53.1	95	155.5	95
TabPFN v2.6	20,000	11.4	1134	15.0	127	58.4	95	220.3	95
TabICL	1,000	0.9	1068	13.0	201	52.6	154	153.7	152
TabICL	20,000	8.2	7755	17.7	1358	55.0	460	206.2	235
Mitra v2	1,000	0.4	965	1.9	367	4.5	334	11.2	327
Mitra v2	20,000	12.9	13224	6.7	1166	10.1	546	18.1	390

K Radius scaling with table size

Table 14 holds rows per shard fixed and grows the table. Proposition 1 bounds the partition radius by $\lfloor m/2 \rfloor$ and the certified k-NN radius by $\lfloor k/2 \rfloor$. The difference is what each bound is tied to. Fix the rows a base learner needs per shard and m grows with N, so the partition radius does too: at 125 rows per shard it is 1.5 at $N = 500$, 7.5 at 2,000 and 31 at 8,000, at 89–94% clean accuracy throughout, and half that at 250 rows per shard (91–95%). Fix k and the k-NN radius saturates at once; 15-NN sits near 6 for every N. Raising k buys radius, but 151-NN pays 8–16 points of accuracy at small N to reach 59 at $N = 8{,}000$, and its radius is still capped at 75 however large the table becomes. A deployer who expects the support table to grow gets a certificate that grows with it from the partition and not from k-NN. All four in-context learners, Mitra v2 included, scale identically. In a deployment m is fixed before the table is seen (A1), so this is the radius a deployment designed for a given table size obtains, not a per-request adaptation.

L Certified partitioning at 40,000 rows

The benchmark caps support at 10,000 rows, while the argument of §6 is about growth: the partition's radius can grow with the number of shards a table affords, certified k-NN's is capped at $\lfloor k/2 \rfloor$. We therefore ran the three largest tables at support sizes $N \in \{5{,}000, 10{,}000, 20{,}000, 40{,}000\}$ where the table has the rows (`magic` stops at 10,000), with TabPFN v2 and Mitra v2 at $m \in \{8, \ldots, 512\}$, shards sorted by digest and label, seed 0 and 500 queries. On the held-out protocol of Table 2 (40/60 split), the partition chooses learner and m on validation, certified k-NN chooses $k \in \{5, \ldots, 1601\}$ and metric, and certified accuracy is read on the 300 test queries.

Table 15 and Figure 6 give the margins. On `adult` the partition trails certified k-NN at 5,000 rows beyond radius 4 (-7.0 points at radius 16, -6.7 under arbitrary edits) and leads at 40,000 ($+3.0$ and

Table 14: Median certified radius as the support table grows, six tables. The partition arms hold 125 or 250 rows per shard so that m grows with N; the certified k-NN arms hold k fixed. Rows per shard is fixed here only to show scaling; a deployment fixes m before the table is seen (A1).

	median certified radius at $N =$				
	500	1,000	2,000	4,000	8,000
partition, 125 rows per shard ($m = N/125$)					
TabPFN v2	1.5	3	7.5	15	31
TabPFN v2.6	1.5	3	7	15	31
TabICL	1.5	3	7	15	31
Mitra v2	1.5	3	7	15	31
partition, 250 rows per shard ($m = N/250$)					
TabPFN v2	0.5	1.5	3.5	7	15
TabPFN v2.6	0.5	1.5	3.5	7	15
TabICL	0.5	1.5	3.5	7	15
Mitra v2	0.5	1.5	3.5	7.2	15
certified k-NN, k fixed					
15-NN	5.5	5.5	5.5	5.5	6
51-NN	14.5	16.5	20.5	20	20
151-NN	37.5	37	46	56.5	59

$+1.7$); on `magic` the radius-32 margin turns from -2.3 to $+2.3$ between 5,000 and 10,000 rows; on `shuttle`, where both methods exceed 97%, the partition leads at every radius and under both threats at 40,000 rows. The trend is the one the structural argument predicts. These are single-seed cells with 300 test queries each, so a single margin carries about ± 2.5 points of sampling noise; we read the direction across N, not individual cells.

Table 15: Held-out margin in certified accuracy (points) of the partition (best of TabPFN v2 and Mitra v2, chosen on validation) over certified k-NN, by support size N. ‡The selected m or k is the top of its grid, so the cell may be truncated.

		ins./del./rel.				any edit		
table	N	$\rho = 4$	$\rho = 16$	$\rho = 32$	$\rho = 64$	$\rho = 8$	$\rho = 16$	$\rho = 32$
`adult`	5,000	-1.3	-7.0	-1.7	+0.0‡	-7.7	-6.7	+0.0‡
`adult`	10,000	+1.3	-1.0	-3.0	-1.3‡	-2.0	-4.3	-2.3‡
`adult`	20,000	+4.0	-0.3	+0.7‡	+0.3‡	+0.7	-0.7	-2.3‡
`adult`	40,000	+2.3	+3.0	+1.0‡	+0.0	+5.0	+1.7	-2.7‡
`magic`	5,000	+6.3	+1.0	-2.3	+0.3‡	+0.3	-8.0	-2.7‡
`magic`	10,000	+7.0	+3.0	+2.3	-1.3	+4.0	+0.0	-6.0
`shuttle`	5,000	+1.3	+1.0	+5.0	-0.3	+0.3	+0.3	-8.3
`shuttle`	10,000	+0.0	-0.7	+2.0	+1.7‡	-0.7	-0.7	-0.3‡
`shuttle`	20,000	+0.3	+1.0	+2.7	+3.3‡	+0.7	+1.0	+1.7‡
`shuttle`	40,000	+0.3	+1.3	+1.0	+2.0	+1.3	+1.0	+1.0

M REGIME ANALYSIS

Table 16 and Figure 7 give the correlations and the margins by radius. With one observation per table nothing is significant after correction, so we read the table descriptively. The margin tends to grow with support size ($+0.31$ to $+0.40$ at radii 4–16, uncorrected p down to 0.08), as the structural ceiling predicts: radius ρ needs $m \geq 2\rho$ shards, so a larger table affords a given radius at a larger shard size. Duplicate-row and majority-class fractions trend negative, and categorical fraction is flat. Treating each backend's row as a separate observation would multiply the sample size by seven and produce stars that do not survive; we report the null result.

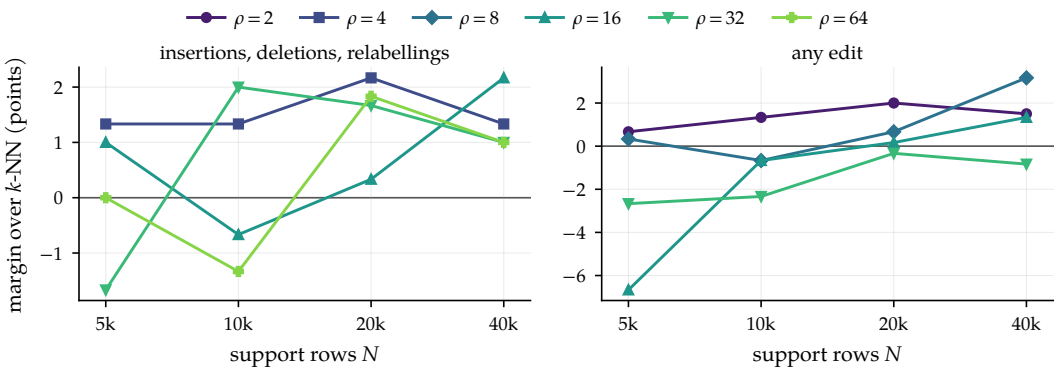


Figure 6: Margin over certified k-NN against support size, median over the tables available at each N (three up to 10,000 rows, two beyond), against insertions, deletions and relabellings (left) and any edit (right). Colour and marker identify the radius ρ in both panels.

Table 16: Spearman correlation between the per-table margin of the in-context partition over certified k-NN (oracle protocol, insertion threat) and table properties, one observation per table ($n = 20$). Holm-corrected over the 25 tests, none is significant (* would mark $p < 0.05$); the last row gives the range of uncorrected p values.

Table property	$\rho = 1$	$\rho = 2$	$\rho = 4$	$\rho = 8$	$\rho = 16$
support size N	+0.08	+0.15	+0.31	+0.34	+0.40
majority-class fraction	-0.11	-0.13	-0.31	-0.22	-0.22
dimension d	+0.11	+0.07	+0.18	+0.23	+0.27
duplicate-row fraction	-0.21	-0.21	-0.34	-0.26	-0.27
categorical fraction	-0.07	-0.10	-0.09	-0.09	-0.03
raw p range	0.37–0.76	0.38–0.76	0.14–0.72	0.15–0.72	0.08–0.90

N LLM DIAGNOSTICS

Table 17: Six open LLMs on SST-2 with 512 demonstrations, 200 queries and three seeds. Learning curve: accuracy of one prompt holding 4 or 256 demonstrations. Undefended: one prompt over all 512, capped at 4,096 tokens (about 200 demonstrations survive the cap). Certified accuracy holds against any r inserted, deleted or relabelled demonstrations. Law MAE: mean absolute error of Proposition 3 over shard counts and radii below the structural ceiling (Table 20 includes the ceiling), in points. Wall: time at $m = 32$ over time at $m = 1$.

	learning curve		undefended	certified (%)		law	wall
model	4 demos	256 demos	$m = 1$	$r = 4, m = 16$	$r = 8, m = 32$	MAE	×
Qwen3.5-2B	88.3	89.9	90.0	87.8	89.0	1.7	8.8
Qwen3.5-4B	90.5	93.3	93.7	86.7	88.0	1.7	2.4
Qwen3.5-9B	93.0	94.2	95.5	91.0	91.5	1.1	2.6
Llama-3.1-8B-It	92.9	91.8	92.3	89.0	88.7	1.2	3.6
Gemma-4-E4B-It	67.5	90.4	90.5	79.5	71.5	4.0	7.7
Gemma-4-12B-It	51.7	56.9	53.7	15.8	16.3	8.5	4.0

Figure 8 shows why the two regimes differ. Gemma-4-12B-It failed to load on the first attempt because the installed Transformers release did not recognise its architecture; the rerun after an upgrade loaded it and produced the numbers in Table 17. Its accuracy is 49.6–54.2% with 4 demonstrations and 52.1–64.9% at every larger context on its learning curve, across seeds, on a task its sibling Gemma-4-E4B-It scores 90% on under the same driver. We did not diagnose the cause. The rule

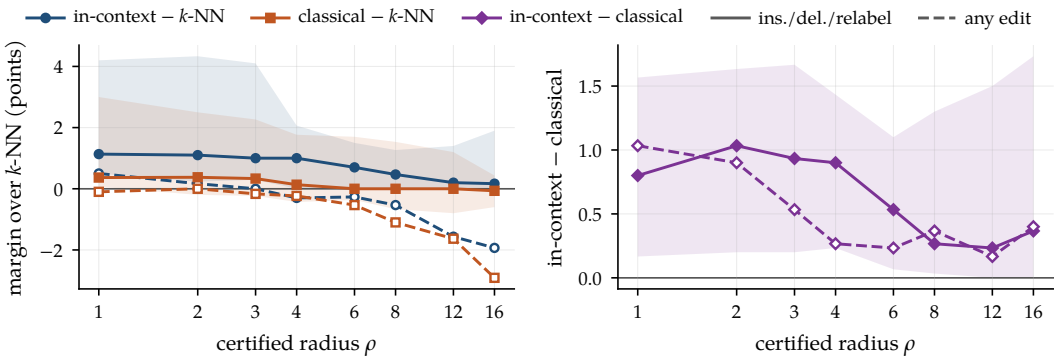


Figure 7: **Left:** median margin over certified k-NN against radius (oracle protocol) for the in-context and classical partitions, against insertions, deletions and relabellings (solid, filled markers, bootstrap 95% band) and against any edit (dashed, hollow markers). **Right:** in-context minus classical partition under the identical certificate, same encoding.

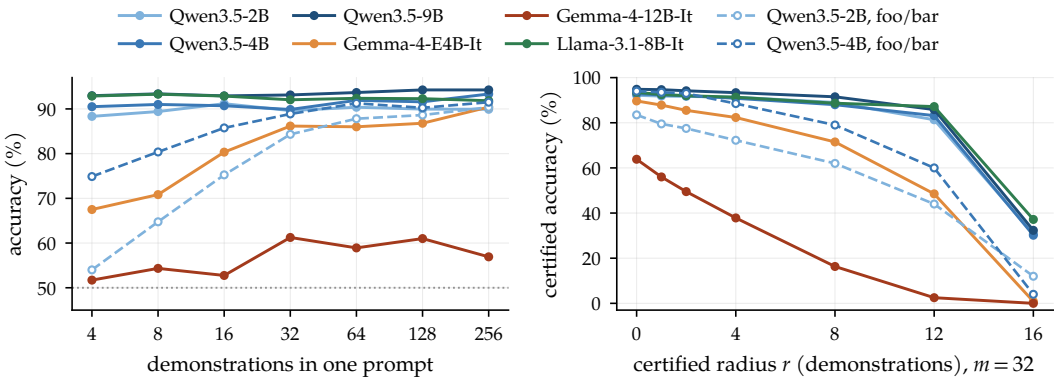


Figure 8: **Left:** learning curves. Four models are flat above 88%, Gemma-4-E4B-It rises 23 points, Gemma-4-12B-It sits near the 50% chance line. Dashed, hollow markers: Qwen3.5-2B and Qwen3.5-4B (seed 0) with the label words `foo`/`bar`, which start lower and are learned from the demonstrations. **Right:** certified accuracy against radius at $m = 32$. The flat models lose under six points by radius 8; the rising one loses 18, the cost of depending on the demonstrations.

we apply is that a model whose undefended accuracy is within 5 points of chance is reported but kept out of the headline, because it has no clean accuracy for certification to spend; the certificate is a statement about votes, and a base learner at chance produces votes worth nothing to certify. The row remains in Table 18.

Table 18: Qwen3.5-2B on SST-2 (natural labels): certified accuracy (%) by shard count and radius, mean over three seeds. Dashes mark radii the shard count cannot express. Prompts were capped at 4,096 tokens, so the $m \leq 2$ prompts hold about 200 of the 512 demonstrations.

m	demos/shard	acc. (%)	med. r	$r = 0$	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 12$	$r = 16$
1	512	90.0	0	90.0	–	–	–	–	–	–
2	256	89.5	0	89.5	42.0	–	–	–	–	–
4	128	91.3	1	91.3	86.5	39.3	–	–	–	–
8	64	91.8	3	91.8	89.3	87.2	34.8	–	–	–
16	32	92.8	7	92.8	91.3	90.5	87.8	31.7	–	–
32	16	92.2	15	92.2	91.8	91.7	91.3	89.0	81.3	30.8

Semantically unrelated labels. Table 19 repeats the Qwen3.5-2B run with the label words replaced by `foo` (negative) and `bar` (positive), following Wei et al. (2023), and with the token cap raised to 16,384 so that no prompt is truncated. With four demonstrations the model is near chance, so every point of accuracy above 50% is learned from demonstrations the adversary may poison. Qwen3.5-4B (seed 0, $m \in \{1, 16, 32\}$) needs fewer demonstrations, 74.9% with four and 91.5% with 256, and pays less for the certificate: 92.0% undefended, 82.0% certified at radius 4 with $m = 16$ (88.5% with $m = 32$) and 79.0% at radius 8 with $m = 32$, where the law predicts 81.3% and 74.8% from its curve. A flatter learning curve means a cheaper certificate, and the law reads which case applies from the curve alone.

Table 19: Qwen3.5-2B on SST-2 with labels `foo`/`bar`: certified accuracy (%) by shard count and radius, mean over seeds 0 and 1 for $m \in \{1, 16, 32\}$ and seed 0 otherwise; the last row is the learning curve (accuracy of one prompt with n demonstrations, both seeds). Dashes mark radii the shard count cannot express.

m	demos/shard	acc. (%)	med. r	$r = 0$	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 12$	$r = 16$
1	512	91.3	0	91.3	–	–	–	–	–	–
2	256	86.0	0	86.0	41.0	–	–	–	–	–
4	128	88.5	1	88.5	81.0	37.5	–	–	–	–
8	64	89.0	3	89.0	87.5	79.5	36.5	–	–	–
16	32	89.5	7	89.5	87.0	84.8	77.5	25.5	–	–
32	16	83.5	11	83.5	79.5	77.5	72.2	62.0	44.0	12.0
learning curve: 4: 54.0, 8: 64.8, 16: 75.2, 32: 84.3, 64: 87.8, 128: 88.6, 256: 90.4										

O ADDITIONAL LAW DIAGNOSTICS

Table 20: Left: error of Proposition 3 against observation over 5,652 cells, restricted to radii the shard count can structurally express. The floor is the error attributable to estimating p_q from 16 subsample repeats; excess is the law's own misspecification. Right: how often the shard count chosen from the learning curve alone matches the observed best.

	law error (points)				design rule	
Base learner	MAE	floor	excess	bias	within 1 pt	exact
In-context						
TabPFN v2	0.59	0.31	0.23	+0.00	95%	74%
TabPFN v2.6	0.60	0.32	0.21	-0.00	90%	70%
TabPFN v3	0.63	0.33	0.22	-0.01	92%	70%
TabICL	0.65	0.33	0.21	+0.00	88%	63%
TabICL v2	0.64	0.34	0.20	-0.02	91%	60%
Mitra	0.60	0.31	0.22	+0.01	91%	68%
Mitra v2	0.62	0.31	0.22	+0.02	90%	68%
Classical						
LogReg	0.89	0.38	0.34	+0.08	79%	46%
RF	0.74	0.41	0.31	+0.00	86%	62%
XGBoost	0.96	0.54	0.33	-0.02	84%	58%
MLP	1.04	0.57	0.40	-0.02	82%	62%
15-NN	0.77	0.30	0.30	+0.00	91%	77%

Table 20 averages the error over the radii each cell can express, ceiling included, and reports the median over cells. Figure 9 (left) instead pools the 54,900 individual $(\text{cell}, \text{radius})$ predictions, which gives a median absolute error of 0.62 points and a mean of 1.78; the gap between median and mean comes from the ceiling radius, where the independence assumption is at its weakest.

Error grows with the shard count, from 0.26 points at $m = 2$ to 2.60 at $m = 256$, but so does the estimation floor, from 0.09 to 1.28, so the law's own misspecification stays near 0.2 points up to

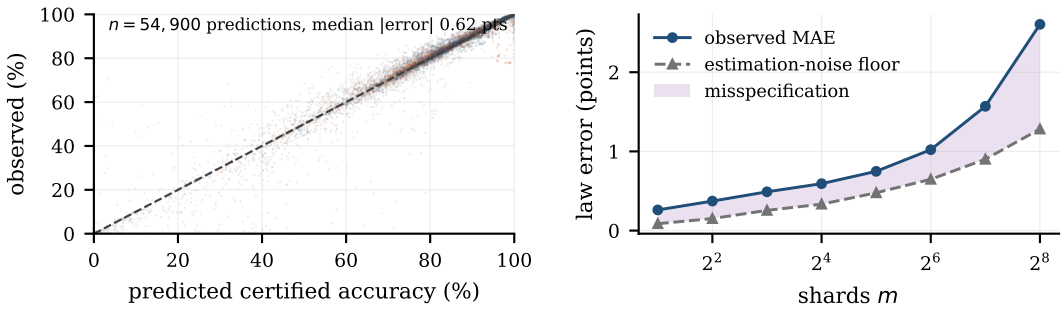


Figure 9: **Left:** predicted against observed certified accuracy, all in-context (blue) and classical (orange) cells. **Right:** the law's error splits into estimation noise in p_q and genuine misspecification; the second stays near 0.2 points until m exceeds 64.

$m = 64$ and reaches 1.27 points only at $m = 256$. Raising the number of learning-curve repeats lowers the floor and would separate the two components further; at $R = 16$ the floor is roughly half the observed error for in-context learners.

P TABPFN V3.5

We also ran TabPFN v3.5, the newest TabPFN release, through the same benchmark: all 20 tables, the full shard grid and its learning curves. Table 21 places it beside the other in-context learners, each evaluated on the same data split, and it follows the rest of the panel closely. Its undefended accuracy, 96.9%, is level with TabPFN v3 and inside the panel's range of 94.9–97.4%. Certification costs it 1.2, 2.9 and 6.9 points at $m = 16$, 64 and 256, inside the range of the other in-context learners at every shard count, and it keeps 92.1% certified accuracy at radius 4 and 90.8% at radius 8 (the others: 91.9–92.7% and 89.5–91.7%). The learning-curve law predicts its certified accuracy to within 0.56 points, as it does for the others (0.53–0.66). Adding it to the in-context arm changes the margin over certified k-NN by at most 0.1 points, and it is the best learner on two to four of the 20 tables depending on the radius. It stands apart only in compute: at $m = 64$ its defended path takes 51× its undefended per-query time, against 3–42× for the other in-context learners.

Table 21: TabPFN v3.5 beside the other in-context learners, all on the same data split of the 20 tables (medians over tables). Acc.: undefended accuracy. Cost: accuracy lost to partitioning at m shards. Certified accuracy against r insertions, deletions or relabellings, best m per table. Law error: mean absolute error of Proposition 3 in points. Time: defended over undefended per-query time at $m = 64$.

		cost (points)			certified acc.		law	time
learner	acc. (%)	$m = 16$	64	256	$r = 4$	$r = 8$	error	$m = 64$
TabPFN v2	96.0	1.1	2.7	7.2	92.0	90.2	0.58	16×
TabPFN v2.6	96.4	1.5	3.3	6.7	91.9	90.7	0.60	24×
TabPFN v3	96.9	1.1	2.9	6.9	91.9	90.4	0.60	42×
TabICL	95.8	1.5	3.5	7.5	92.0	90.8	0.66	21×
TabICL v2	96.4	0.9	2.1	4.4	92.7	91.7	0.58	34×
Mitra	94.9	0.2	1.9	6.2	92.4	89.5	0.64	3×
Mitra v2	97.4	1.0	3.5	8.1	92.4	90.3	0.53	3×
TabPFN v3.5	96.9	1.2	2.9	6.9	92.1	90.8	0.56	51×