SUPPORT-COMPILED FEATURE FOLDING: MORE EVIDENCE AT LOWER MEMORY ACROSS TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Wide tables offer tabular foundation models more evidence, but accessing it can exhaust their memory: full-width pairwise mixing grows quadratically with the number of columns, while feature selection makes inputs affordable by discarding evidence. We ask whether using more features requires interacting over all of them at once. We introduce Support-Compiled Feature Folding (SCFF), a training-free inference framework that encodes wide tables through bounded calls to a frozen backbone. SCFF organizes support-ranked features into a strong Core and a candidate Tail, folds them into narrow feature groups, and support-checks the Tail's added evidence before a single contextual prediction. This converts quadratic feature-interaction work into linear-in-width work with a bounded local working set, without ensembling predictions or training new parameters.

On the exhaustive 18-dataset wide-table slice of fixed AMLB-29, TabZilla, and TabArena snapshots, SCFF improves dataset-macro accuracy and NLL on all six evaluated backbones. All four matched-width comparisons retain favorable 95% dataset-bootstrap intervals on locked folds, with relative error reductions up to 26.1%. Median paired GPU-memory savings are 2.09–$2.36\times$, and the ratio of separately observed maximum peaks reaches $34.3\times$. Under a measured peak-memory ceiling, SCFF uses the saved budget to preserve more support-selected evidence, improving accuracy by 4.06 and 3.72 points over the widest feasible single leaf on predeclared wide-Core strata of TabICLv2 and TabPFN-3.

1 INTRODUCTION

Tabular foundation models promise to turn labeled examples into a predictor without task-specific training (Müller et al., 2022; Hollmann et al., 2025; Qu et al., 2025). On a wide table, however, even presenting the available evidence can be expensive. Thousands of columns offer many possible clues, but their usefulness is uneven: some carry a clear signal, others matter only together, and many are redundant. Full-width feature attention pays for their pairwise interactions before the model can reason across examples. Feature selection makes this affordable by discarding columns, including any useful evidence beyond its cutoff. The resulting dilemma is concrete: *how much evidence must a frozen model give up simply to fit the table in memory?*

Consider *robert*, with 7,200 features. In an illustrative all-feature folding pilot, TabICLv2 uses 59.3 GiB of peak allocated GPU memory. Processing the same columns in groups of at most 128 lowers the peak to 1.83 GiB while keeping the checkpoint, context encoder, and prediction head unchanged (Figure 1). This gap motivates a different question: *does using a wide table require interacting over its full width at once?*

The opportunity lies at an existing boundary inside the model. In feature-interaction ICL architectures, a feature encoder compresses each row into a fixed-dimensional representation; the context encoder then reasons across rows after the raw feature axis has disappeared. The costly pairwise mixing, with $\Theta(TD^2)$ interactions for T rows and D features, occurs before this boundary. Our key insight is that *the feature encoder can see a narrow group while the context encoder still sees every row*. Multiple groups can contribute to the same row representation, allowing the downstream model to make one prediction from their combined evidence.

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.

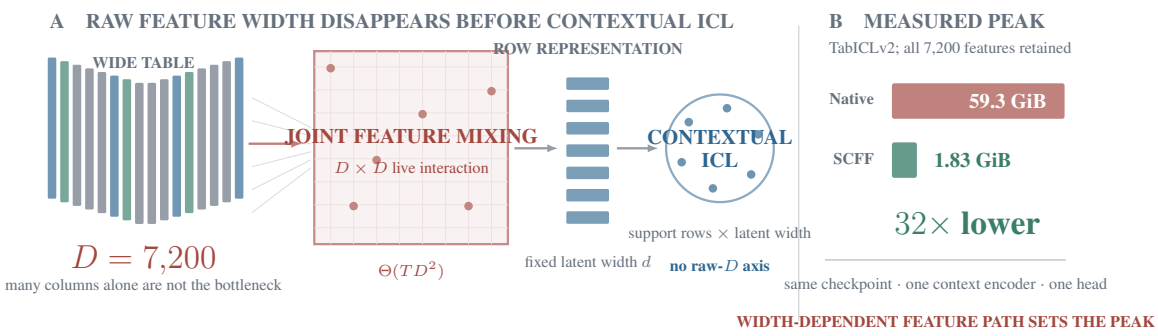


Figure 1: **Feature width is a distinct bottleneck in tabular ICL. A:** Width-dependent interactions precede the fixed-dimensional row representation consumed by contextual ICL. **B:** An illustrative TabICLv2 folding pilot retains all 7,200 features of *robert* in leaves of at most 128 columns, reducing peak allocated GPU memory from 59.3 to 1.83 GiB with the same downstream stack and head. Table 1 evaluates the final SCFF method separately.

We turn this observation into *Support-Compiled Feature Folding* (SCFF). Support labels organize the columns into a Core of individually informative features and a candidate Tail. Each is divided into bounded groups, or *leaves*, processed by the same frozen feature encoder. SCFF merges their row representations before the context encoder. With fixed leaf capacity, feature-interaction work grows linearly with total width, while all support examples remain available to the final prediction.

Making evidence cheaper to encode is only half the problem: adding more evidence can also add redundancy and noise. SCFF therefore gives the Tail a specific role—to contribute information that the Core has not already captured. It removes the Core-aligned component and uses reciprocal tests on support subsets to check whether the residual contributes reproducible evidence. An accepted residual augments the Core; rejection preserves its representation exactly. Every column is encoded, but Tail inclusion is conditional. Binary tasks use this support-checked update; multiclass tasks use Bounded Core. A token-preserving LimiX adapter extends the design to a different model interface.

The practical payoff is more than fitting a wide input. Memory spent on feature interactions competes with the budget for retaining features and providing labeled context. Just as retrieval improves the value of context slots (Thomas et al., 2024), folding can improve how the model spends its feature-side budget. We test this on a fixed OpenML Wide-Table benchmark: all eligible dataset versions in the AMLB-29, TabZilla, and TabArena snapshots with at least 128 raw features, under declared class and development-exclusion rules. Its 18 datasets let us ask whether the design transfers across frozen models, whether folding helps beyond selection, and whether the saved memory can buy useful evidence. Our contributions are:

- **A training-free width compiler.** We introduce SCFF, which combines support-conditioned routing, bounded feature folding, and support-checked residual evidence before one context-encoder and label-head pass. It trains no parameters and does not ensemble predictions.
- **Broad gains on wide tables.** Across six frozen backbones on the exhaustive 18-dataset wide-table slice, SCFF improves every dataset-macro accuracy and NLL comparison. Relative error falls by up to 26.1%, relative NLL by up to 19.0%, and the maximum-peak memory ratio reaches 34.3×. On locked folds, all four matched-width backbones retain favorable 95% dataset-bootstrap intervals for both predictive metrics.
- **A mechanism for reinvesting memory in evidence.** Holding the selected Core fixed, bounded leaves reduce peak memory across three latent-state backbones. Under the measured folded-memory ceiling, SCFF keeps more selected features than the widest feasible single leaf and gains 4.06 and 3.72 accuracy points on the predeclared wide-Core strata of TabICLv2 and TabPFN-3. Support-checked Tail recovery then tests evidence outside the Core before the same single prediction.

2 Related Work

Learning a predictor, adapting its input. Prior-data fitted networks amortize Bayesian-style inference over synthetic tasks, predicting from a labeled support table without dataset-specific gradient updates (Müller et al., 2022). TabPFN and its successors improve the prior, tokenizer, and inference stack (Hollmann et al., 2025; Grinsztajn et al., 2025; 2026), while TabICL scales the paradigm toward larger tables (Qu et al., 2025; 2026). Released systems such as LimiX, EXAONE Tabular, and TabFM broaden the family of pretrained tabular predictors (Zhang et al., 2025; Eo et al., 2026; Google Research, 2026). These advances make a frozen predictor a useful starting point. SCFF asks how much wider an input it can use through inference-time organization alone, exploiting the interface between feature encoding and contextual prediction.

Making more evidence affordable. Retrieval allocates finite context to relevant support examples (Thomas et al., 2024). We address the complementary feature-width budget. Classical filters rank or delete columns using support statistics (Guyon & Elisseeff, 2003), with multiple-testing control providing a principled high-evidence subset (Benjamini & Hochberg, 1995); feature engineering instead changes the input schema (Tschalzev et al., 2026). Test-time divide-and-conquer extends TabPFNv2 without retraining (Ye et al., 2025), while TabPFN-Wide uses continued pre-training to handle extreme feature counts (Kolberg et al., 2025). SCFF explores a complementary choice: retain the frozen predictor and reorganize its feature computation. It uses ANOVA/BH to identify a Core, encodes the remaining columns as a candidate Tail, and combines intermediate representations before one contextual prediction. This locates the width adaptation inside feature encoding, where the pairwise cost arises.

Bounded computation, shared prediction. Deep Sets and Set Transformer learn permutation-invariant aggregation end to end (Zaheer et al., 2017; Lee et al., 2019). SCFF instead reuses one feature encoder on disjoint leaves and combines its row representations without a learned set encoder. This resembles the bounded-interface principle of hierarchical classification (Silla & Freitas, 2011; Qu et al., 2025), but feature groups produce row representations rather than predictions. They are merged before the shared context encoder, which retains all support and query rows, and the label head executes once. The relevant design principle is to expand the evidence handled by a model through repeated use of a bounded interface. For features, the shared object is the representation on which the model will reason about the task.

3 Support-Compiled Feature Folding

The design begins with a separation: a model can collect evidence in narrow groups and still reason over all examples together. SCFF uses the boundary between feature encoding and contextual prediction to implement this separation. For an episode (X_s, y_s, X_q) with D features and T support-plus-query rows, write

$$f_\theta(X_s, y_s, X_q) = H_\theta(C_\theta(F_\theta([X_s; X_q], y_s); y_s)), \tag{1}$$

where F_θ is the width-dependent feature encoder, C_θ is the context encoder, and H_θ is the label head. We keep all pretrained parameters fixed. SCFF changes how F_θ is executed and leaves the remaining model layers unchanged.

3.1 Organize evidence before compressing it

Treating all columns equally would mix strong individual signals with weak or redundant evidence before either is understood. Algorithm 1 first ranks columns by one-way ANOVA: class separation relative to within-class variation on a median-filled support copy. The Benjamini–Hochberg discovery count K determines Core size; the strongest $\max(1, K)$ columns form $\mathcal{C}$, and the rest form the candidate Tail $\mathcal{T}$. Weak individual scores need not imply useless joint information, so the Tail is encoded for a later check. Routing reads support labels only; the model retains its native preprocessing and missing-value handling.

Within each stratum, round-robin assignment creates the minimum number of disjoint leaves of at most $M = 128$ columns. Every preprocessed feature occurs in exactly one leaf. The same feature

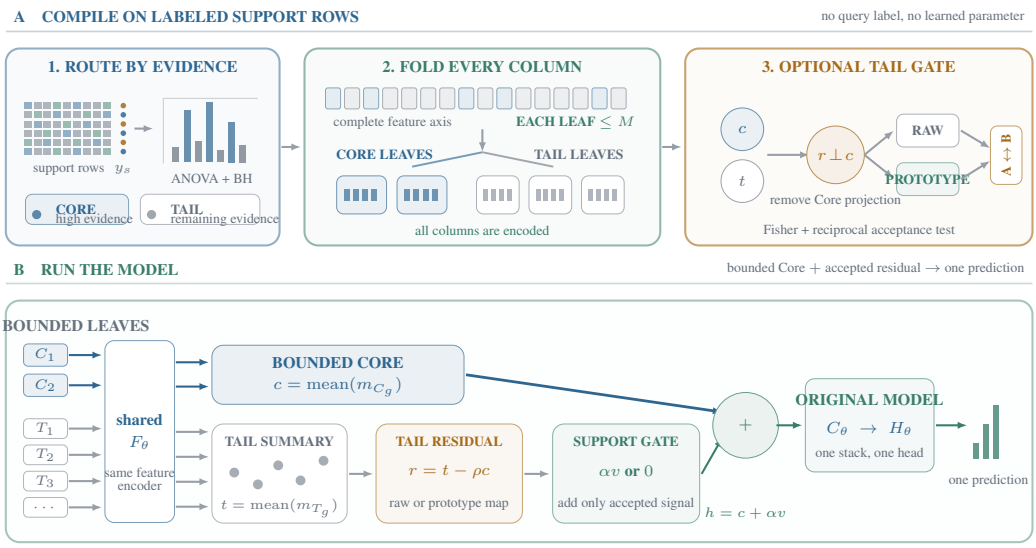


Figure 2: **SCFF folds the Core and optionally recovers Tail evidence.** Support labels route every column to Core or Tail, and the same feature encoder processes each bounded-width leaf. Core leaves capture the strongest support evidence without full-width competition. Tail leaves propose nonredundant evidence, which enters the prediction only after a reciprocal support-subset acceptance test. The combined row representation enters the original context encoder and label head once.

Algorithm 1 Support-Compiled Feature Folding (row-representation adapter)

Input: episode (X_s, y_s, X_q); model $(F_\theta, C_\theta, H_\theta)$; leaf capacity M **Output:** one query distribution

ROUTE — organize every column

Score columns by support-only ANOVA
$K \leftarrow$ BH discovery count at FDR 0.05
$\mathcal{C} \leftarrow$ top $\max(1, K)$; $\mathcal{T} \leftarrow [D] \setminus \mathcal{C}$
Partition each stratum round-robin into
  disjoint leaves P_g, $|P_g| \leq M$

FOLD + ENCODE — reuse the feature encoder

for each Core or Tail leaf P_g **do**
  $m_g \leftarrow F_\theta([X_s; X_q]_{P_g}, y_s)$
end for

CORE — preserve strong evidence

$c \leftarrow |\mathcal{G}_C|^{-1} \sum_{g \in \mathcal{G}_C} m_g$

OPTIONAL TAIL — test added information

$\alpha \leftarrow 0$; $v \leftarrow 0$
if $\mathcal{T} \neq \varnothing$ and the task is binary **then**
  Average Tail messages and center on support
  Remove the Core-aligned projection
  Form normalized raw and prototype candidates
  Split every support class into halves A, B
  for each candidate v' **do**
    Fit Fisher steps on A and B
    Test each step on the opposite half
    Require NLL and separation gains on A, B
    Require no accuracy loss on A, B
  end for
  Select the accepted update with the largest worst-direction NLL gain
  For one Tail leaf, allow Fisher-only validation
end if

PREDICT — use the original context encoder

return $H_\theta(C_\theta(c + \alpha v; y_s))$

Contract: all features are encoded once; query labels are never read; C_θ and H_θ each execute once. Rejection sets $\alpha = 0$ and returns the exact Core representation.

encoder maps each leaf to a row-aligned representation

$$m_g = F_\theta([X_s; X_q]_{P_g}, y_s) \in \mathbb{R}^{T \times d_{\text{ICL}}}. \tag{2}$$

Each leaf therefore preserves every support and query row in the same latent coordinates. This is the point where many narrow feature computations become compatible inputs to one contextual prediction.

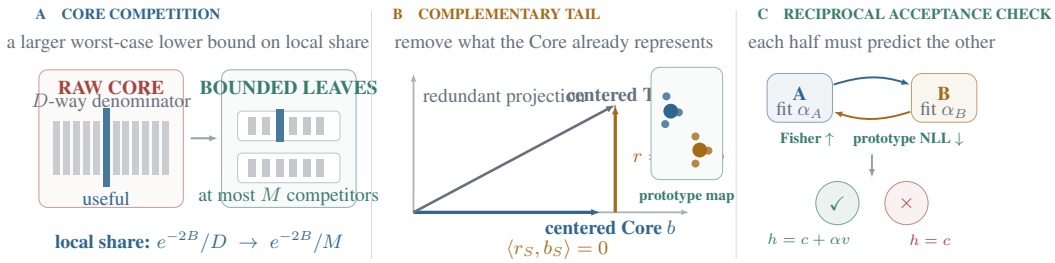


Figure 3: **Why bounded folding can improve prediction.** Bounded leaves increase local share; the optional Tail path removes Core-aligned content and accepts only support-consistent residual directions.

3.2 Combine strong evidence, test what the Tail adds

The Core provides the initial representation. SCFF averages its leaf messages into $c = |\mathcal{G}_C|^{-1} \sum_{g \in \mathcal{G}_C} m_g$. This lets the feature encoder model rich interactions among at most M strong features at a time, then combines the resulting row representations at the model's standard interface.

The remaining question is whether the Tail adds information beyond this representation. SCFF averages its leaf messages, removes the global direction already aligned with the centered Core representation, and considers both the resulting raw direction and a support-prototype projection. The prototype map is support-fitted: query rows use all-support class centroids, while each support row uses a leave-one-out own-class centroid. A class-balanced split then fits a Fisher step on each half and tests that step on the opposite half; reciprocal nearest-prototype checks evaluate the fixed candidate in both directions. Because the candidate prototype uses the full support set before these checks, this is an acceptance test, not independent cross-fitting or a separate validation set. The candidate must improve both checks; otherwise $\alpha = 0$ and SCFF returns the exact Core representation. Multiclass episodes also use this fallback. The resulting representation

$$h = c + \alpha v, \qquad p(y_q \mid X_s, y_s, X_q) = H_\theta(C_\theta(h; y_s)) \tag{3}$$

produces one complete query distribution. Prototype scores test a candidate direction; they are never averaged model predictions. Appendix A gives the projection, prototype map, reciprocal support-subset test, and full derivations.

3.3 Why the change can help

Bounded competition changes the inductive bias. If one feature-attention normalization has logits in $[-B, B]$, a feature among K competitors receives weight at least e^{-2B}/K. Full-width execution therefore gives a bound of e^{-2B}/D, while a leaf of width at most M gives e^{-2B}/M. The lower bound on local share increases by D/M before leaf aggregation. This motivates a predictive hypothesis: bounded groups can give useful features more local influence. Aggregation can attenuate or cancel that change, so matched Raw-Core and Bounded-Core controls measure its net effect.

Additional evidence must earn its place. After projection, the raw Tail residual is orthogonal to the centered Core on the support rows. Reciprocal subset checks then require its class evidence to transfer in both directions. If that evidence is absent, the update is zero and the combined representation is exactly the Core representation. The held-out Tail controls test whether support-accepted evidence also improves unseen queries; Appendix A states the corresponding transfer condition.

The expensive feature interaction becomes width-linear. For a pairwise feature mixer, a width-D feature encoder costs $\Theta(TD^2d)$. With $G \simeq D/M$ bounded leaves, the total becomes

$$G\,\Theta(TM^2d) = \Theta(TDMd), \tag{4}$$

while sequential execution keeps an $O(TM^2)$ feature-interaction working set. The G representations are combined into one $T \times d_{\mathrm{ICL}}$ representation, so the context encoder and label head still run once at the original context length.

3.4 Reusing the same idea across architectures

SCFF requires compatible row representations before a shared context encoder and must reproduce native inference. TabPFN-3, TabICLv1/v2, TabPFNv1, and RefineICL-24L (Anonymous, 2026) expose Equation (1). LimiX keeps a feature-token axis; its adapter retains Core slots and folds the remainder into bounded Tail slots with support-RMS rescaling (Appendix B.2).

4 Evaluating the Evidence–Memory Tradeoff

The empirical question is whether making a wide table cheaper to encode also makes its evidence more useful. We first compare SCFF with native inference, then hold the selected features fixed to isolate folding, and finally let the memory savings pay for additional evidence. These comparisons distinguish the benefit of choosing features from the benefit of processing them differently.

Benchmark: 18 wide-table datasets. Our main evaluation uses **18 classification datasets from AMLB-29, the released TabZilla collection, and TabArena v0.1**, spanning 128–7,200 raw features. These comprise all eligible dataset versions in the fixed registry snapshot: at least 128 features, 2–50 classes, no missing target, and no semantic duplicate. Registry metadata fixes this population before any model outcome is read. We use fold 0 of these datasets for the explicitly described finite method selection and reserve folds 1–4 for locked confirmation. Confirmation uses new query rows on the same datasets. The dataset list and benchmark membership appear in Appendix Table 10.

Models and evaluation. We evaluate five released backbones and RefineICL-24L, a frozen checkpoint from a concurrent anonymous submission (Anonymous, 2026), over five fixed folds. Each native–SCFF pair uses the same checkpoint, preprocessing, split, and query rows. We average folds within datasets, then macro-average datasets, reporting paired point estimates for accuracy, NLL, time, and peak GPU memory. Paired dataset-bootstrap intervals use these within-dataset averages. Four backbones have matched-width comparisons; LimiX tests a token-aligned adapter, and TabPFNv1 tests a capacity extension. Coverage and adapter contracts appear separately in Appendix Table 8. We use open-source benchmark data and model checkpoints, and provide our reproduction code in the supplementary material.

Freezing the design before confirmation. Fold 0 selects the parameter-free Tail update, prioritizing the number of compatible backbones improving both metrics, then mean NLL and accuracy. We lock the method and configuration before folds 1–4. The complete-method confirmation and the incremental Tail contrast remain separate: the latter requires at least three of five compatible backbones to improve both metrics. All five are reported in Appendix Table 20; the complete-method intervals appear in Table 12. Deterministic stratified query folds are disjoint and support sets are their complements. Seeds, caps, checkpoints, and execution details appear in Appendix Table 11.

Separating selection, folding, and recovery. *Raw Core* sends the ranked Core through one standard feature-encoder call; *Bounded Core* folds the same columns into leaves of at most $M = 128$. *Tail Off* sets $\alpha = 0$; *Tail On* adds the selected Tail update. These pairs ask whether folding improves an already selected feature set and whether the Tail contributes beyond it. Tail On encodes every feature once; rejection reproduces Tail Off.

Testing the design choices. Table 3 tests seven fixed TabICLv2 alternatives on the same 18 datasets and five folds: random routing, no Core projection, and a sequence of Tail variants that adds the residual, prototype map, Fisher step, reciprocal support-subset acceptance, single-leaf fallback, and binary-task restriction. Every variant retains all features in one prediction; deltas are full SCFF minus each complete variant, not additive effects. Random routing preserves leaf sizes and coverage, while the projection control changes only the residual geometry.

Turning memory savings into evidence. We compare Bounded Core with the widest support-ranked single leaf fitting its measured peak-memory budget; memory alone selects the prefix before quality is compared. Separate interventions vary leaf width, add up to 512 label-independent columns, and increase nested support context on fixed query rows. An earlier all-feature mass-Huber variant additionally tests context reinvestment under Native@2,048's measured budget (Appendix E).

5 RESULTS

5.1 WIDE-TABLE QUALITY IMPROVES WITH A SMALLER MEMORY FOOTPRINT

Table 1: **Main results on the fixed 18-dataset wide-table registry from AMLB-29, TabZilla, and TabArena.** The panel spans 128–7,200 raw features. We compare SCFF with native inference over five fixed folds within each backbone's native evaluation contract.

Backbone	ΔAcc. pp ↑	Error ↓	ΔNLL ↓	NLL ↓	Median E2E ↑	Median mem. ↓
TabPFN-3	+1.874	+11.8%	-0.05065	+11.8%	1.00×	2.15×
TabICLv1	+5.419	+26.1%	-0.09099	+18.3%	0.62×	2.36×
TabICLv2	+1.564	+11.1%	-0.03557	+9.6%	0.56×	2.32×
TabPFNv1‡	+3.071	+13.9%	-0.04341	+8.5%	–	–
LimiX-2M†	+0.700	+4.7%	-0.01178	+3.2%	1.72×	2.46×
RefineICL-24L	+3.730	+22.0%	-0.08223	+19.0%	4.84×	2.09×

Quality is dataset-macro; resources are native/SCFF medians over verified paired folds and E2E includes support compilation. Error and NLL are relative reductions. Model-specific native eligibility, ‡capacity and †token contracts, and full resource distributions are detailed in SI, Appendix Tables 8 and 15.

The first result is that the memory reduction need not come at the expense of predictive quality. SCFF improves accuracy and NLL on all six evaluated backbones. On the dataset-macro aggregate, relative error falls by up to 26.1% and relative NLL by up to 19.0%. All four matched-width rows have favorable locked-fold dataset-bootstrap intervals for both metrics (Appendix Table 12); the separately reported capacity and token-interface extensions cross zero. Across the four matched-width backbones, fold-paired median memory savings are 2.09–$2.36\times$; ratios of the largest observed native and SCFF peaks reach 5.2–$34.3\times$. Median end-to-end speed is near parity for TabPFN-3, slower for both TabICL backbones, and faster for LimiX and RefineICL. Contracts and intervals appear in Appendix Tables 8 and 14. Leave-one-dataset-out effects remain favorable for all matched-width backbones (Appendix Table 13); paired resource distributions appear in Appendix Table 15. The next comparisons ask how much of this improvement comes from selection and how much from the way the selected evidence is processed.

5.2 BOUNDED CORE AND TAIL RECOVERY CONTRIBUTE DISTINCT GAINS

Table 2: **Selection supplies quality; folding supplies capacity.** Panel A decomposes native → selected Core → bounded Core on 18 datasets and five folds; Panel B adds optional Tail evidence.

Backbone	Selection − native ΔAcc. pp / ΔNLL	Bounded − selection ΔAcc. pp / ΔNLL	Raw/Bounded max. peaks ↑
A. Full-panel Core decomposition (18 datasets, five folds)			
TabPFN-3	+1.563 -0.03414	+0.233 -0.00639	30.1×
TabICLv1	+5.175 -0.07959	+0.162 -0.00911	32.3×
TabICLv2	+0.690 -0.01576	+0.739 -0.01683	34.0×
B. Optional Tail extraction: Tail On − Tail Off (held-out folds 1–4)			
TabPFN-3	+0.107 -0.00985	–	–
TabICLv1	+0.080 -0.00234	–	–
TabICLv2	+0.145 -0.00294	–	–

Cells report ΔAccuracy pp / ΔNLL. Selection supplies most quality; folding adds a smaller gain on all three backbones. The 30.1–34.0× ratios divide the separate largest raw and bounded Core peaks. Appendix Tables 16 and 20 report intervals and all Tail-compatible models.

Feature selection alone already helps. Across the three backbones, it identifies a strong Core ($+0.690$ to $+5.175$ accuracy points), and folding that same Core improves both predictive metrics again. Thus the selected set does not fully explain the gains. The corresponding ratio between the largest observed raw and bounded Core peaks is 30.1–$34.0\times$. Both all-five-fold folding intervals exclude zero on TabICLv2. A secondary folds 1–4 same-Core analysis also finds both intervals favorable on

TabPFN-3; TabICLv1 remains uncertain (Appendix Table 17). The Core need not be the end of the evidence budget. On held-out folds, Tail On improves both point estimates on TabPFN-3, TabICLv1, and TabICLv2 while encoding every remaining column and retaining exact Core fallback. Appendix Tables 16 and 20 give the full intervals and all five compatible models.

5.3 How evidence is organized matters

Splitting a table is not sufficient by itself. The following controls keep coverage and prediction count fixed while changing which features share a role and how additional evidence enters the representation.

Table 3: **Component controls around full SCFF.** Each row reports a paired TabICLv2 point estimate, full SCFF minus one fixed alternative; positive accuracy and negative NLL favor SCFF.

Ablated variant	ΔAcc. pp ↑	ΔNLL ↓
Random Core/Tail assignment	$+1.878$	-0.03811
Centered Tail (no Core projection)	$+0.054$	-0.00125
Raw residual only	$+0.018$	-0.00403
Fixed prototype update	$+0.586$	-0.01680
Fisher-selected update	$+0.513$	-0.01481
Validated update, no single-leaf fallback	$+0.209$	-0.00874
Validated update, no binary-task restriction	$+0.202$	-0.00881

18 datasets × five folds; every variant retains all features and produces one prediction. The first two change one component; the last five build the Tail update step by step. Table 2 compares the complete Core and Tail paths across backbones.

Core allocation. Random routing preserves leaf sizes, coverage, and inference count, yet falls below native by 0.314 accuracy points and 0.00254 NLL; full SCFF is 1.878 points and 0.03811 NLL better. The grouping must therefore reflect the task's support evidence. Removing Core projection costs 0.054 points and 0.00125 NLL, consistent with the value of separating new from redundant evidence.

Tail acceptance. Raw residual nearly matches accuracy but worsens NLL by 0.00403; fixed-prototype and Fisher-selected updates trail SCFF by $0.586/0.01680$ and $0.513/0.01481$ accuracy points/NLL. Reciprocal support-subset checking gives the best point estimates among these fixed controls.

Keeping the evidence fixed. The same-K comparison provides the resource counterpart to the Core quality control. Across 30 paired units, bounded leaves use 13.4% of the single-leaf incremental peak memory at $1.41\times$ latency. The same selected evidence has become substantially cheaper in memory. Section 5.5 tests whether spending this saving on additional features improves prediction.

5.4 Moderate leaf width preserves useful interaction capacity

Table 4: **Moderate leaves preserve interactions.** All columns retained; four datasets, five folds.

M	Median leaves	ΔAcc. pp ↑	ΔNLL ↓	Median time (s)	Peak GPU (GiB)
32	40.5	-0.498	$+0.00872$	3.135	1.986
64	21.0	$+0.287$	-0.01310	2.666	1.100
128	11.0	$\mathbf{+0.255}$	$\mathbf{-0.01070}$	3.281	1.666

Deltas are versus native; peak is the maximum over 20 runs. Fold 0 selected $M = 128$.

The leaf cap balances affordable interactions against the information lost when related features are separated. $M = 32$ loses quality, whereas 64 and 128 improve over native. The full evaluation retains the fold-0-selected $M = 128$. The benefit comes from bounded but expressive groups, rather than making the groups as small as possible.

5.5 Memory savings buy more useful evidence

The practical test is what a model can do with the released memory budget. We examine two uses: accommodating more examples and retaining a wider Core.

Table 5: **Width determines how SCFF responds to more context.** Each cell is Δaccuracy points / ΔNLL relative to native inference on the same nested support prefix and fixed query rows.

Dataset	Raw D	Context 128	Context 512	Context 2,048
kddcup09_appetency	212	$+0.000/-0.00791$	$+0.000/+0.00408$	$+0.000/-0.00166$
philippine	308	$-4.102/+0.08142$	$-2.246/+0.00761$	$+0.879/-0.01000$
dilbert	2,000	$+5.762/-0.18864$	$+2.148/-0.01944$	$-0.098/+0.00049$
guillermo	4,296	$+2.344/-0.08459$	$+8.594/-0.10163$	$+8.887/-0.11406$

The four datasets are fixed from raw-width bands before evaluation. Appendix Table 25 reports memory, time, and the complete protocol for all 12 cells.

At 4,296 features, the gain over native rises from 2.344 to 8.887 accuracy points as both arms receive more context. The 212-feature task stays near native, with mixed intermediate responses. Across all four, peak memory falls from 37.6% to 27.0% of native as context grows from 128 to 2,048 rows. This motivates a stricter comparison: require the single-leaf baseline to fit within the folded route's measured memory ceiling.

On the predeclared $K > 128$ strata, folding retains a median 244 versus 96 features on TabICLv2, gaining 4.06 accuracy points and reducing NLL by 0.100 across 10 datasets. The same rule gains 3.72 points and reduces NLL by 0.086 on nine TabPFN-3 datasets. Figure 4 connects the extra retained evidence to these gains and their 95% intervals.

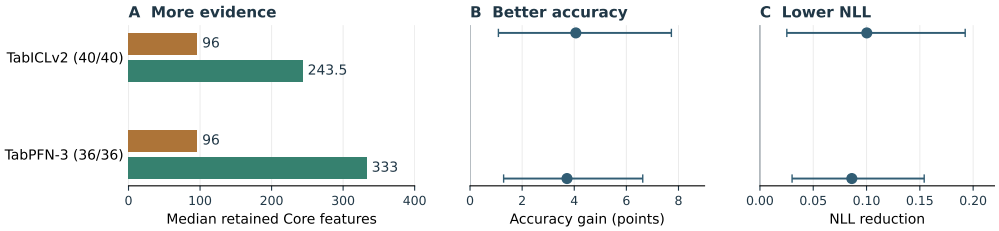


Figure 4: **More evidence below a measured peak-memory ceiling improves prediction.** Predeclared $K > 128$ datasets: TabICLv2 (10), TabPFN-3 (9), four folds each. The widest memory-feasible prefix from a fixed grid (ochre) stays below the paired folded peak (green) in all 40/40 and 36/36 folds; folded/prefix peak ratios are 1.17× (median) and 1.13× (mean). Accuracy/NLL effects show dataset-bootstrap 95% intervals. Both routes make one full prediction.

Table 6: **Equal-memory gains extend beyond the highest-width strata.** Each comparison uses one frozen model member and four held-out folds per dataset. The single leaf is the widest support-ranked prefix below the paired folded peak in the fixed width grid; intervals resample datasets.

Backbone	Scope	Data	Folds	ΔAcc. pp [95% CI]	NLL reduction [95% CI]
TabICLv2	Full registry†	18	72	$+2.28$ $[+0.50, +4.63]$	0.0560 [0.0112, 0.1149]
	Strict complete cases	16	64	$+2.56$ $[+0.60, +5.16]$	0.0630 [0.0134, 0.1288]
	Core width $K > 128$	10	40	$+4.06$ $[+1.09, +7.73]$	0.1002 [0.0252, 0.1926]
TabPFN-3	Full registry	18	72	$+1.86$ $[+0.47, +3.63]$	0.0431 [0.0110, 0.0841]
	Core width $K > 128$	9	36	$+3.72$ $[+1.29, +6.63]$	0.0862 [0.0302, 0.1541]

†Six TabICLv2 one-leaf identity folds have equal predictions but a 0.19–1.39% difference in one-shot peak readings. Full-registry includes these zero-effect pairs; strict complete cases retain the 16 datasets with all four folds within budget. Per-dataset results and protocols appear in Appendix Tables 22 and 23.

Both backbones also improve across the complete registry. Since memory alone fixes each prefix before quality is measured, this comparison completes the argument: saved memory buys more evidence for the same frozen predictor.

Conclusion. Using more features need not require interacting over all of them at once. SCFF improves six frozen backbones through bounded feature computations and one contextual prediction. Its memory savings can preserve more useful evidence, turning width adaptation into an inference-time opportunity for tabular foundation models.

REFERENCES

Anonymous. What does tabular icl compute? in-situ representation refinement through attention-gated updates, 2026. Concurrent submission to ICLR 2027.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x. URL https://academic.oup.com/jrsssb/article/57/1/289/7035855.

Moonjung Eo, Min-Kook Suh, Hye-Seung Cho, Jiwon Kim, Seoyoon Kim, Sangjun Nam, and Soonyoung Lee. EXAONE Tabular 1.0: Technical report. *arXiv preprint arXiv:2608.25774*, 2026. URL https://arxiv.org/abs/2608.25774.

Google Research. TabFM: Tabular foundation models. Software repository, 2026. URL https://github.com/google-research/tabfm. Repository accessed September 18, 2026; no technical report was available.

Léo Grinsztajn, Klemens Flöge, Oscar Key, et al. TabPFN-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025. URL https://arxiv.org/abs/2511.08667.

Léo Grinsztajn, Klemens Flöge, Oscar Key, et al. TabPFN-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026. URL https://arxiv.org/abs/2605.13986.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. URL https://www.jmlr.org/papers/v3/guyon03a.html.

Noah Hollmann, Samuel Müller, Lennart Purucker, et al. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Christopher Kolberg, Jules Kreuer, Jonas Huurdeman, Sofiane Ouaari, Katharina Eggensperger, and Nico Pfeifer. TabPFN-Wide: Continued pre-training for extreme feature counts. *arXiv preprint arXiv:2510.06162*, 2025. URL https://arxiv.org/abs/2510.06162.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=u3vyZf5Jv2.

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. URL https://proceedings.mlr.press/v267/qu25d.html.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026. URL https://arxiv.org/abs/2602.11139.

Carlos N. Silla, Jr. and Alex A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1–2):31–72, 2011. doi: 10.1007/s10618-010-0175-9. URL https://doi.org/10.1007/s10618-010-0175-9.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval & fine-tuning for in-context tabular models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c40daf14d7a6469e65116507c21faeb7-Abstract-Conference.html.

Andrej Tschalzev, Nick Erickson, Yuyang Wang, Huzefa Rangwala, Stefan Lüdtke, Heiner Stuckenschmidt, and Christian Bartelt. TabPrep: Closing the feature engineering gap in tabular benchmarks. *arXiv preprint arXiv:2606.02384*, 2026. URL https://arxiv.org/abs/2606.02384.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at TabPFN v2: Understanding its strengths and extending its capabilities. *arXiv preprint arXiv:2502.17361*, 2025. URL https://arxiv.org/abs/2502.17361.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, et al. LimiX: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025. URL https://arxiv.org/abs/2509.03505.

Reproducibility statement

The primary matched-fold comparisons are indexed by checkpoint hashes, data manifests, fold records, adapter revisions, and matched native–SCFF runs. Appendix C specifies the inference and statistical contracts. The supplementary archive includes code, manifests, paired fold-level metric records and table renderers for the primary locked-fold evaluation, and its failure histories. It does not include per-query predictions or model weights.

AI use statement

Generative AI tools were used to assist with code implementation, debugging, experiment orchestration, and manuscript editing. The authors defined the research questions and protocols, reviewed all generated code and prose, and verified reported results against executable tests and hash-audited artifacts.

A Formal Properties and Scope

The central design claim is that a wide input can be processed through bounded feature interactions and still supply one contextual prediction. We first establish how this changes computation, then explain how the Tail is tested for information beyond the Core. This connects the resource argument to the predictive hypothesis: coverage and bounded interactions follow from the construction, while better predictions depend on the frozen representation and the transfer of support evidence to unseen rows.

A.1 Feature coverage and resource scaling

Folding would be a trivial memory reduction if it simply dropped columns. The first property rules out that explanation: the computational units get narrower while the encoded feature set remains complete.

Within Core and Tail, SCFF stably ranks columns, distributes them round-robin over the minimum number of width-M leaves, and restores original column order inside each leaf. For the resulting leaves $P_1, \ldots, P_G$,

$$P_g \cap P_{g'} = \varnothing \ (g \neq g'), \qquad \bigcup_{g=1}^{G} P_g = [D], \qquad 1 \leq |P_g| \leq M. \tag{5}$$

Equation (5) follows because the two strata are disjoint and exhaustive and the distribution changes neither membership nor multiplicity. Consequently, every preprocessed feature is encoded exactly once. This guarantees feature coverage; it does not reconstruct the nonlinear full-width representation.

This changes the competition faced by each feature before the row representations are combined. Suppose one feature-attention row has logits $a_1, \ldots, a_K \in [-B, B]$. For any index j,

$$\mathrm{softmax}(a)_j = \frac{e^{a_j}}{\sum_{k=1}^{K} e^{a_k}} \geq \frac{e^{-B}}{K e^{B}} = \frac{e^{-2B}}{K}. \tag{6}$$

A width-D interaction therefore has lower bound e^{-2B}/D, whereas a width-M leaf has lower bound e^{-2B}/M. Their ratio is D/M (or $|\mathcal{C}|/M$ in the Raw-Core comparison). This compares worst-case local lower bounds; it does not bound the ratio of realized attention weights or prediction errors because folding changes the logits and removes cross-leaf interactions.

The resource benefit comes from replacing one large interaction map with a sequence of small ones. For a pairwise feature mixer, one leaf costs $O(TM^2d)$. With $G = O(D/M)$ leaves, total feature-interaction work is $O(TDMd)$ and sequential execution bounds the live interaction map by $O(TM^2)$. The final combined representation remains $T \times d_{\mathrm{ICL}}$, but total device memory is not constant in D: the present adapters can retain $O(GTd_{\mathrm{ICL}})$ leaf representations in addition to weights, contextual activations, and runtime workspace. We therefore treat the measured peak-memory results as empirical systems evidence rather than as an equality with the asymptotic interaction working set.

A.2 Tail residual and reciprocal support-subset acceptance

Encoding all columns solves access to evidence; it does not decide how much of that evidence should affect a prediction. A useful Tail update should add something beyond the Core and retain class information across support subsets. The projection and acceptance test implement these two requirements.

Let $t = |\mathcal{G}_T|^{-1} \sum_{g \in \mathcal{G}_T} m_g$ be the averaged Tail message and $\mathrm{ctr}(z) = z - |S|^{-1} \sum_{i \in S} z_i$ center a representation by its support rows. Write $b = \mathrm{ctr}(c)$ and $u = \mathrm{ctr}(t)$. SCFF removes the global Tail direction aligned with the Core:

$$\rho = \frac{\langle u_S, b_S \rangle}{\|b_S\|_2^2}, \qquad r = u - \rho b. \tag{7}$$

Flatten the centered support representations b_S and u_S. For nondegenerate b_S, Equation (7) gives

$$\langle r_S, b_S \rangle = \langle u_S, b_S \rangle - \frac{\langle u_S, b_S \rangle}{\|b_S\|_2^2} \langle b_S, b_S \rangle = 0. \tag{8}$$

If the denominator is degenerate, the implementation sets the projection coefficient to zero. Orthogonality therefore holds for the raw residual in the nondegenerate case; it is not a claim of statistical independence. The prototype-mapped candidate lies in the convex hull of the centered class prototypes and hence in their span, but the nonlinear map need not preserve the raw residual's orthogonality.

The raw residual can still contain variation unrelated to the task. The second candidate concentrates it toward support class prototypes. For class centroids μ_k and support class frequencies q_k, define

$$p_k(r_i) = \frac{\exp(-\|r_i - \mu_k\|_2^2/s)}{\sum_{\ell} \exp(-\|r_i - \mu_\ell\|_2^2/s)}, \qquad r_i^{\mathrm{proto}} = \sum_k p_k(r_i) \left(\mu_k - \sum_{\ell} q_\ell \mu_\ell \right), \tag{9}$$

where s is the median positive squared distance between class centroids. For a support row, its own-class distance uses a leave-one-out centroid. This is a support-fitted candidate, not an independently cross-fitted representation: query rows use centroids from all support rows. Cross-half separation and nearest-prototype checks below accept or reject the fixed candidate and its proposed step without reading query labels. Both the raw and prototype-mapped candidates are normalized to the support RMS of b, so the update coefficient has the same scale across episodes.

For support subset Q, write the Fisher ratio of $b + \alpha v$ as

$$J_Q(\alpha; v) = \frac{B_{0,Q} + 2\alpha B_{1,Q} + \alpha^2 B_{2,Q}}{W_{0,Q} + 2\alpha W_{1,Q} + \alpha^2 W_{2,Q}}. \tag{10}$$

The stationary points are the real roots in $[0, 1]$ of a quadratic, together with the two endpoints, so the finite candidate set gives the exact maximizer used by the method. After splitting each class alternately into A and B, each half proposes a step and SCFF uses

$$\alpha_v = \min(\widehat{\alpha}_A, \widehat{\alpha}_B). \tag{11}$$

An accepted multi-leaf update must improve Fisher separation and nearest-prototype NLL in both directions without lowering either support accuracy. It therefore satisfies

$$J_A(\alpha_v; v) > J_A(0; v), \qquad J_B(\alpha_v; v) > J_B(0; v). \tag{12}$$

If no candidate passes, $\alpha = 0$ and Equation (3) returns the Bounded Core representation exactly. The single-leaf Tail fallback uses the two-way Fisher condition alone because no cross-leaf Tail merge is required; unlike the multi-leaf update, it does not guarantee prototype-NLL improvement. Multiclass tasks also take the exact Core fallback.

The accepted direction is added before the original context encoder:

$$h = c + \alpha v, \qquad p(y_q \mid X_s, y_s, X_q) = H_\theta(C_\theta(h; y_s)). \tag{13}$$

This expression also makes the exact fallback explicit: setting $\alpha = 0$ returns $h = c$ while preserving one call to the context encoder and label head.

A.3 A CONDITIONAL SUPPORT-TO-QUERY STATEMENT

The remaining link is from a support-side improvement to useful evidence on new rows. The following statement makes that link explicit and identifies the empirical question addressed by the held-out Tail experiment.

Let $\Delta_S(v)$ be the smaller prototype-NLL gain across the two support directions and $\Delta_Q(v)$ the corresponding population gain on unseen query rows. The validation step establishes $\Delta_S(v) > 0$ for an accepted multi-leaf update. To connect this fact to query performance, an additional stability condition is required:

$$|\Delta_Q(v) - \Delta_S(v)| \le \varepsilon. \tag{14}$$

Under this condition, any accepted update with $\Delta_S(v) > \varepsilon$ has $\Delta_Q(v) > 0$. Exchangeability motivates estimating the query gain from support splits but does not itself imply Equation (14), nor does the proposition guarantee improvement after an arbitrary context encoder and head. The held-out Tail experiment in Section D tests precisely this empirical transfer.

Finite implementation checks. The following tests connect the formal objects above to one executable implementation. Their numerical tolerances verify the tested cases; they are not substitutes for the algebraic statements or broad predictive evidence.

Table 7: **Executable checks for the residual/prototype update.**

Check	Setting	Observed
Exact coverage	1025 features	1025 retained once
Bounded feature encoder	9 leaves	maximum width 128
Support-only routing	query representations shifted 100×	validation unchanged
Residual geometry	accepted raw update	$\lvert\langle r_S, b_S\rangle\rvert = 5.1e-07$
Prototype geometry	rank-1 class span	projection error 6.0e-08
Split-half validation	support halves A, B	Fisher $0.080 \to 2.684$
Prototype prediction	both directions	minimum NLL gain 0.439
Nuisance width	2–16 symmetric Tail leaves	maximum Core change 0.0e+00
Exact fallback	rejected binary / multiclass Tail	weights 0/0, bit-exact

B IMPLEMENTATION AND ARCHITECTURE INTERFACES

The framework depends on where a model permits evidence to be combined. This section follows a table through support compilation and the architecture adapter, showing how the bounded-computation idea is realized while preserving the downstream prediction interface.

B.1 SUPPORT COMPILER

The method operates on a median-filled copy of the support features. It computes one-way ANOVA statistics, uses the Benjamini–Hochberg discovery count at nominal FDR 0.05 to determine Core size, and assigns the highest-F_j columns to the Core. At least one column initializes a nonempty Core. ANOVA/BH therefore routes columns; it does not delete the Tail. The model still receives its native preprocessing and missing-value handling. Every leaf uses the same feature encoder and receives all support and query rows. Core and Tail leaf representations are averaged with equal leaf weight before support-only validation is applied.

The compatible insertion point differs across architectures. Table 8 records the comparison and baseline for each reported model; these distinctions prevent a capacity extension from being interpreted as a matched-width efficacy result.

Table 8: **Model-specific evaluation settings.** Matched-width comparisons use the same features, comparisons within native limits respect each model's output constraints, and capacity extensions use wider inputs than the original model.

Model	Comparison	Coverage	Interpretation
TabICLv2	matched full width	90 folds / 18 datasets	direct comparison
TabPFN-3	matched full width	90 folds / 18 datasets	direct comparison
TabICLv1	matched full width	90 folds / 18 datasets	direct comparison
RefineICL-24L	matched full width	90 folds / 18 datasets	direct comparison
TabPFNv1	support-ranked 128 vs. native 100-feature view	85 folds / 17 datasets	capacity extension
TabPFNv2	latent representation interface	standard-path test	representation boundary
LimiX-2M	token-aligned representation combination	85 folds / 17 datasets	LimiX adapter

Table 9: Execution checks for every reported result.

Quantity	Required value
Model checkpoints	1
Feature partitions from support data	1
Context-encoder calls per query	1
Label-head calls per query	1
Averaged final predictions	0
Query labels used to build feature groups	false
Feature budget reduced after OOM	false

B.2 ROW-REPRESENTATION AND TOKEN-PRESERVING ADAPTERS

TabPFN-3, TabICLv1/v2, TabPFNv1, and RefineICL-24L expose compatible row representations before a shared context encoder. Their adapters combine aligned $T \times d_{\mathrm{ICL}}$ messages. TabPFNv1's standard inference uses a fixed 100-feature view, so its 17-dataset comparison asks whether support compilation extends usable capacity rather than matching full input width.

LimiX token adapter. LimiX forms one token from each native group of g scalar columns before its alternating feature–row transformer. Averaging at the row-representation interface used by the other models would destroy this trained token axis. The LimiX variant therefore encodes leaves of at most M columns with the released tokenizer, padding the last incomplete native group of each leaf. Tokens follow compiled Core-first leaf order, with original column order within each leaf. Let $w_{\max} = \max_j |P_j|$ be the actual maximum leaf width. For T_f resulting feature tokens, the output budget is

$$B = \min(T_f, \lceil w_{\max}/g \rceil), \quad B_{\mathrm{C}} = \min(T_f - 1, \max(1, \mathrm{round}(0.75B))), \quad B_{\mathrm{T}} = B - B_{\mathrm{C}}. \tag{15}$$

When $T_f > B$, the first B_{C} Core tokens remain unchanged and the remaining tokens are split evenly across B_{T} Tail slots, averaged, and support-RMS rescaled. For the reported LimiX-2M experiments,

$g = 2$ and $M = 512$; 192 Core and 64 Tail slots result only when $B = 256$. Smaller leaves use correspondingly smaller budgets. When $T_f \leq B$, all encoded tokens are retained. A one-leaf plan bypasses compilation and calls native inference exactly. This token-preserving combination is training-free and single-prediction, but it does not use the split-half row-representation Tail update analyzed above.

B.3 Verifying one complete prediction

Repeated feature-encoder calls are essential to folding. The following checks ensure that they collectively supply one prediction and that the measured savings do not arise from lost columns or altered evaluation data.

For every reported evaluation, sorted leaf indices must equal $0, \ldots, D-1$ without duplication; each leaf must respect its width bound; and the plan and Tail validation must read support features and labels only. A rejected Tail update must reproduce the exact Bounded Core representation. Each query uses one checkpoint member, one feature partition, one context-encoder call, one label-head call, and one complete class distribution. Leaf-local prototype scores select a candidate Tail update and are never combined as final predictions. Query labels are read only after the predictions have been produced.

C Data and Statistical Protocol

To test width adaptation, the evaluation population must be defined by width before observing model gains. To test the method itself, native and folded inference must then see the same task. The registry and pairing rules below establish these two conditions; the locked-fold analyses assess whether the selected design continues to help on new query rows.

C.1 Benchmark selection

The OpenML Wide-Table panel is the exhaustive set of dataset versions in the fixed AMLB-29, released TabZilla, and TabArena v0.1 snapshots satisfying the eligibility rule, which uses input metadata only. Exact OpenML data IDs identify versions and semantic duplicates are excluded before outcomes are observed. The dataset population itself is outcome-blind; fold 0 of this fixed population is nevertheless used for the finite method-selection procedure described below. This is a controlled mechanism benchmark assembled from public registries, not an official leaderboard aggregate.

Table 10: **Datasets in the OpenML Wide-Table Holdout.** The population is the exhaustive set of classification datasets in the public fixed registry snapshot from September 4, 2026 covering AMLB-29, the released TabZilla task collection, and TabArena v0.1 that satisfy the preregistered metadata rule: at least 128 raw features, 2–50 classes, no missing target, and no exact data ID or semantic reissue from the earlier exploratory development set. Fold 0 of this fixed panel was then used for the declared finite method selection; folds 1–4 were reserved for locked-method evaluation. OpenML data IDs identify the immutable dataset versions; no model prediction or query label enters dataset selection. "TabZilla list" denotes the paper's released full task collection, while 379 and 457 denote the corresponding OpenML benchmark suites.

OpenML data ID	Dataset version	Benchmark source	Rows	Features	Classes
5	arrhythmia	TabZilla list	452	279	13
312	scene	TabZilla list	2,407	299	2
1036	sylva_agnostic	TabZilla list	14,395	216	2
1038	gina_agnostic	TabZilla list	3,468	970	2
1116	musk	TabZilla list	6,598	169	2
1476	gas-drift	TabZilla list	13,910	128	6
1477	gas-drift-different-concentrations	TabZilla list	13,910	129	6
41142	christine	TabZilla list	5,418	1,636	2
41143	jasmine	AMLB-29; TabZilla 379/list	2,984	144	2
41144	madeline	AMLB-29	3,140	259	2
41145	philippine	AMLB-29; TabZilla list	5,832	308	2
41159	guillermo	TabZilla 379/list	20,000	4,296	2
41161	riccardo	TabZilla list	20,000	4,296	2
41163	dilbert	TabZilla list	10,000	2,000	5
41164	fabert	TabZilla list	8,237	800	7
41165	robert	TabZilla list	10,000	7,200	10
46933	hiva_agnostic	TabArena v0.1 / suite 457	3,845	1,617	3
46939	kddcup09_appetency	TabArena v0.1 / suite 457	50,000	212	2

Table 11: **Concrete locked-fold reproduction contract.** Values are read from the released driver and per-fold receipts; hash prefixes identify the exact frozen artifacts.

Item	Frozen value
Population	18 listed OpenML versions (17 for TabPFNv1); selected from registry metadata before model outcomes
Fold construction	deterministic stratified 5-fold; split seed 20260904; fold 0 development, folds 1–4 locked confirmation
Overlap	query folds are pairwise disjoint; each support set is the complement and therefore overlaps other support sets
Caps and sampling	support ≤ 2048, query ≤ 1024; stratified cap seed derived per dataset; query subsampling seed 20260905
Method	ANOVA/BH FDR 0.05; $M = 128$; balanced round-robin Core/Tail leaves; final residual/prototype gate
Execution	one member, one preprocessing view, one context-stack call and one label-head call; no prediction averaging
OOM policy	halve the paired query chunk on both paths only; never reduce support rows or features; exclude incomplete pairs
Hardware and memory	PPU-ZW810 (72 SM, CUDA-compatible); both arms paired on the same device; peak allocated bytes from PyTorch
Precision	released model-native inference precision; native and SCFF share the same checkpoint, runtime and preprocessing
TabPFN-3	checkpoint `d0d865d54dfb`; source `b03a4f9547bd`; temperature 0.9
TabPFN-3 view	one full-width `quantile_uni`/numeric transform; no feature or class shift, fingerprint, polynomial expansion, or within-registry feature subsampling
TabICLv1	checkpoint `04c5c1d261c1`; source `59a957cd644b`; temperature 0.9
TabICLv2	checkpoint `bdc7dbd5e4ff`; source `59a957cd644b`; temperature 0.9
RefineICL-24L	step-15000 checkpoint `071de368d16d`; source `2d0bb39ce7fa`; temperature 0.9
TabPFNv1 extension	checkpoint `3c9aadaeddbf`; source `a9298ae0fe2a`; temperature 0.8
Protocol hashes	manifest `e9339459ecb8`; evaluation `fdbf4d9345d2`; statistics `56d8255b4a95`

The five query folds form a partition of each dataset under deterministic stratified splitting. Consequently, fold 0 query rows never occur as query rows in folds 1–4, although they enter those folds' support sets, as standard cross-validation requires. The confirmation therefore tests new query rows after freezing the method, not transfer to unseen datasets. Dataset-bootstrap intervals estimate variation across the fixed dataset population; they do not turn this protocol into leave-dataset-out method development.

C.2 Pairing, aggregation, and resources

A paired evaluation fixes dataset version, fold, support/query indices, preprocessing, checkpoint, query chunking, and runtime. Both methods must complete without fallback or replacement. For metric m, we first average fold-level paired effects within dataset d and then macro-average dataset effects,

$$\Delta m = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \left(\frac{1}{|\mathcal{F}_d|} \sum_{f \in \mathcal{F}_d} [m_{d,f}^{\mathrm{SCFF}} - m_{d,f}^{\mathrm{native}}] \right). \tag{16}$$

The headline table contains fold 0 as well as locked folds 1–4; its point estimates therefore describe the fixed five-fold panel, not an untouched test of method selection. We therefore confirm the complete method separately on folds 1–4 and exclude fold 0 from that analysis. Unless an interval is printed, the table reports a point estimate rather than statistical significance.

We additionally resample datasets with replacement after the within-dataset fold average, preserving the dataset as the unit of generalization. All four matched-width comparisons have favorable locked-fold 95% intervals for both metrics. The token-aligned comparison is not reconstructed under this fold contract; the capacity extension remains a positive point estimate whose interval crosses zero.

Table 12: **Locked-fold confirmation of the complete SCFF method.** The method selected on fold 0 is evaluated only on folds 1–4. Effects are averaged within dataset before 100,000 paired dataset-bootstrap resamples. ‡ denotes the separately declared capacity extension.

Backbone	Data	ΔAccuracy pp [95% CI]	ΔNLL [95% CI]
TabPFN-3	18	+1.956 [+0.212, +4.808]	-0.05103 [-0.11702, -0.00816]
TabICLv1	18	+5.348 [+1.663, +9.846]	-0.08828 [-0.15570, -0.03260]
TabICLv2	18	+1.615 [+0.546, +2.999]	-0.03603 [-0.06176, -0.01520]
RefineICL-24L	18	+3.846 [+0.323, +9.136]	-0.08394 [-0.17533, -0.01588]
TabPFNv1‡	17	+3.063 [-0.962, +7.368]	-0.04479 [-0.11306, +0.02135]

Positive averages also invite a distributional question: does the conclusion depend on one exceptional dataset? The leave-one-dataset-out analysis below removes each dataset in turn from the locked-fold comparison. Favorable effects remain on all four matched-width backbones, complementing the dataset-bootstrap intervals with a direct check of individual influence.

Table 13: **Locked-fold gains survive every single-dataset deletion.** For each matched-width backbone, we remove each dataset in turn and report the least favorable remaining macro effect for each metric. The named dataset is the deletion that produces that worst case.

Backbone	Full ΔAcc.	Min LODO ΔAcc.	Full ΔNLL	Max LODO ΔNLL
TabPFN-3	+1.956	+0.689 (fabert)	-0.05103	-0.02276 (fabert)
TabICLv1	+5.348	+3.831 (madeline)	-0.08828	-0.06831 (fabert)
TabICLv2	+1.615	+1.133 (guillermo)	-0.03603	-0.02690 (robert)
RefineICL-24L	+3.846	+1.595 (riccardo)	-0.08394	-0.04775 (riccardo)

LODO: leave one dataset out. Positive Accuracy and negative NLL favor SCFF.

Table 14: **Paired dataset-bootstrap intervals for the headline panel.** Effects are computed after averaging folds within each dataset; brackets are 95% percentile intervals over 100,000 dataset resamples. † is token-aligned and ‡ is a capacity extension; the remaining four rows are matched-width comparisons.

Backbone	Datasets	ΔAccuracy pp [95% CI]	ΔNLL [95% CI]
TabPFN-3	18	+1.874 [+0.036, +4.806]	-0.05065 [-0.11723, -0.00654]
TabICLv1	18	+5.419 [+1.727, +9.977]	-0.09099 [-0.15981, -0.03462]
TabICLv2	18	+1.564 [+0.528, +2.918]	-0.03557 [-0.06120, -0.01496]
RefineICL-24L	18	+3.730 [+0.380, +8.764]	-0.08223 [-0.16831, -0.01732]
LimiX-2M†	17	+0.700 [-0.769, +2.338]	-0.01178 [-0.03719, +0.01125]
TabPFNv1‡	17	+3.071 [-0.957, +7.297]	-0.04341 [-0.11047, +0.02201]

End-to-end time sums prediction and support compilation over complete units. Peak allocated memory is measured by the PyTorch CUDA allocator. In the text's maximum-peak comparison, each backbone's ratio divides its largest observed native peak by its largest observed SCFF peak; those maxima need not occur on the same fold. It is not the maximum of paired-fold ratios. Feature partitions may be reused for repeated query batches, but the reported end-to-end number charges compilation once.

Table 15: **Paired resource ratios across dataset-fold runs.** Each ratio compares native and SCFF on the same query fold. Memory is native/SCFF peak allocated GPU memory; speed is native/SCFF end-to-end time including support compilation. Medians and 10th–90th percentiles describe folds, not ratios of cross-task extrema; only directly measured or hash-verified native references are included. RefineICL has 90 predictive folds but only 72 resource pairs: all 18 fold-2 native reference file hashes differ from the retained baseline artifact and are excluded rather than imputed.

Backbone	Pairs	Memory, median [P10, P90]	Speed, median [P10, P90]
TabPFN-3	90	2.15× [1.36, 21.99]	1.00× [0.63, 2.04]
TabICLv1	90	2.36× [1.56, 25.39]	0.62× [0.40, 0.96]
TabICLv2	90	2.32× [1.54, 27.29]	0.56× [0.33, 0.93]
RefineICL-24L	72	2.09× [1.23, 23.90]	4.84× [3.13, 6.74]
LimiX-2M	85	2.46× [1.02, 4.84]	1.72× [0.69, 5.61]
TabPFNv1	85	0.89× [0.30, 0.97]	0.18× [0.06, 1.17]

For every result, we record the checkpoint SHA-256, source revision, dataset, fold, support/query hashes, preprocessing, method hash, adapter revision, and runtime. Changing the router, FDR threshold, leaf width, residual map, or validation rule defines a new protocol.

D Mechanism Validation

The main table shows what SCFF achieves as a complete method. The experiments here ask why. We first hold the feature set fixed, then examine where folding changes the representation, and finally test whether weaker evidence can contribute beyond the resulting Core. This progression assigns distinct roles to selection, bounded computation, and Tail recovery.

D.1 Why separate Core folding from Tail recovery?

The headline comparison changes both the representation of selected evidence and the treatment of lower-ranked columns. Two paired controls identify these effects. Raw Core and Bounded Core receive the same ranked feature set; only the latter uses bounded leaves. Tail Off and Tail On share the exact Bounded Core representation; only Tail On executes the remaining features and adds a support-checked Tail residual. Thus the first contrast tests bounded local competition, while the second asks whether evidence outside the Core passes a reciprocal support-subset acceptance test. The component ablation in main Table 3 uses the same paired units. Its coupled controls are not additive effects.

Table 16: **Uncertainty for the full-panel Core decomposition.** Selection-only minus native and Bounded Core minus selection-only over all five folds; brackets are 95% paired dataset-bootstrap intervals after averaging folds within each dataset.

Backbone	Contrast	ΔAcc. pp ↑	ΔNLL ↓
TabPFN-3	Selection − native	+1.563 [-0.197, +4.481]	-0.03414 [-0.09835, +0.00407]
	Bounded − selection	+0.233 [-0.094, +0.730]	-0.00639 [-0.01629, +0.00042]
TabICLv1	Selection − native	+5.175 [+1.465, +9.716]	-0.07959 [-0.14791, -0.02282]
	Bounded − selection	+0.162 [-0.284, +0.683]	-0.00911 [-0.02096, +0.00105]
TabICLv2	Selection − native	+0.690 [-0.155, +1.884]	-0.01576 [-0.03360, -0.00119]
	Bounded − selection	+0.739 [+0.117, +1.656]	-0.01683 [-0.04114, -0.00048]

Selection supplies most of the quality gain. Holding that selected Core fixed, bounded folding improves both point estimates on all three backbones; the ratio of separate maximum Core peaks is 30.1–34.0×, while both folding intervals exclude zero only on TabICLv2. Thus the robust folding claim is capacity preservation with a consistent, smaller quality increment, not that folding explains the full native-to-SCFF gain.

Fold 0 was used for development. Recomputing the fixed-Core comparison on query folds 1–4 alone gives the secondary paired analysis in Table 17. This contrast was examined after the five-fold analysis and uses new query rows from the same datasets, not new datasets. It separates the development fold without changing selected columns, checkpoints, or the two paired prediction routes.

Table 17: **Locked-query-fold same-Core contrast.** Bounded Core minus Raw Core on folds 1–4; both use the same selected columns, checkpoint, support and query indices. Positive accuracy difference and NLL reduction favor bounded folding. Brackets are 95% dataset-cluster bootstrap intervals over 18 datasets, averaging four folds within dataset. This secondary contrast was examined after the all-five-fold analysis; folds are new query rows from the same datasets, not unseen datasets.

Backbone	ΔAcc. pp ↑	NLL reduction ↑
TabPFN-3	+0.308 [+0.005, +0.808]	+0.00744 [+0.00047, +0.01802]
TabICLv1	+0.171 [-0.277, +0.660]	+0.00851 [-0.00283, +0.02134]
TabICLv2	+0.768 [+0.156, +1.702]	+0.01787 [+0.00084, +0.04291]
Equal-model mean	+0.416 [+0.099, +0.842]	+0.01127 [+0.00111, +0.02591]

D.2 When does bounded Core processing change predictions?

The mechanism predicts where a difference should emerge: below one leaf, the two routes should nearly coincide; above two leaf widths, local feature interactions differ substantially. We partition folds by the selected Core width K under the fixed $M = 128$ cap, average paired differences within each dataset, and then macro-average datasets in each bin. Table 18 reports every bin for all three latent-state backbones. At $K > 256$, both quality point estimates favor bounded Core on each backbone; $K \leq 128$ is approximately an identity control. Only five datasets contribute to the widest bin per backbone, however, and this stratification was examined after evaluation. It describes where the observed effect lies, not an independently confirmed width threshold or a guarantee of predictive improvement.

Table 18: **Exploratory same-Core quality by selected width.** Raw and bounded Core share columns, checkpoint, and query fold. Within each width bin, folds are averaged within dataset before dataset-macro aggregation. Positive values favor bounded Core. Bins are defined by the $M = 128$ leaf cap; datasets can contribute different folds to adjacent bins. These descriptive contrasts were examined after the main evaluation, not preregistered as confirmatory tests.

Backbone	Core K	Datasets	Folds	ΔAcc. pp ↑	NLL reduction ↑
TabPFN-3	$K \leq 128$	9	45	+0.000	+0.00000
TabPFN-3	129–256	5	21	-0.316	-0.00268
TabPFN-3	$K > 256$	5	24	+1.084	+0.02545
TabICLv1	$K \leq 128$	10	44	+0.000	+0.00000
TabICLv1	129–256	6	22	+0.137	+0.00178
TabICLv1	$K > 256$	5	24	+0.714	+0.03472
TabICLv2	$K \leq 128$	10	44	+0.000	+0.00000
TabICLv2	129–256	6	22	+1.215	+0.03033
TabICLv2	$K > 256$	5	24	+1.917	+0.04975

D.3 How was the Tail update selected and confirmed?

The Tail presents a harder design problem than the Core: more encoded information may be redundant or may obscure a useful representation. We therefore compare ways of proposing and accepting that information before testing the chosen rule on new query rows. Fold 0 compared four specified Tail updates across all five compatible backbones. Selection first maximized the number of backbones with simultaneous dataset-macro accuracy and NLL improvement, then compared mean NLL and accuracy. This is development evidence over a finite candidate set, not a claim that all possible Tail updates were searched.

Table 19: **Selecting the Tail update across models.** Each variant is evaluated on fold 0 over all five compatible models. Selection first maximizes the number of models with simultaneous accuracy and NLL improvement, then compares mean NLL and accuracy.

Tail update	Passing models	Mean ΔAcc. pp	Mean ΔNLL
Raw residual update	2/5	$+0.090$	-0.00130
Cross-fitted residual update	2/5	$+0.079$	-0.00172
Centered cross-fitted update	2/5	$+0.080$	-0.00163
Full residual/prototype validation	**3/5**	$\mathbf{+0.025}$	$\mathbf{-0.00304}$

Each model effect is a dataset macro over the 18 predeclared wide datasets (17 for TabPFNv1). The selected rule is the only complete variant that improves both metrics on at least three of five models. Its implementation was fixed before folds 1–4 were evaluated.

The selected implementation was fixed before folds 1–4 were evaluated. The confirmation comprises 72 paired units for each 18-dataset backbone and 68 for TabPFNv1's 17-dataset panel within its native limits. Three of five backbones improve both metrics, satisfying the predeclared three-of-five rule. Held-out transfer is not universal: TabPFNv1 loses 0.007 accuracy points despite a negligible NLL gain, and RefineICL-24L degrades both metrics. The five-backbone comparison therefore supports a consistent three-backbone latent-state result plus a documented interface boundary, not universal Tail improvement or backbone-level statistical significance.

Paired dataset-bootstrap intervals quantify effect strength rather than replace the predeclared cross-backbone criterion. TabICLv2 has favorable intervals for both Tail effects: Accuracy $+0.145$ pp $[+0.009, +0.334]$ and NLL -0.00294 $[-0.00673, -0.00012]$; the other backbones' smaller Tail intervals cross zero for at least one metric. Thus simultaneous Accuracy and NLL improvement on all three representative latent-state backbones is the main Tail result, while TabICLv2 provides the strongest dataset-level interval evidence.

Table 20: **Complete Tail-extraction confirmation.** Tail On minus Tail Off over held-out folds 1–4 for every compatible backbone.

Backbone	ΔAcc. pp ↑	ΔNLL ↓
TabPFN-3	+0.107	-0.00985
TabICLv1	+0.080	-0.00234
TabICLv2	+0.145	-0.00294
TabPFNv1	-0.007	-0.00007
RefineICL-24L	-0.299	+0.00559

The selected Tail update improves both primary metrics on three of five compatible models, satisfying the predeclared selection rule.

D.4 What do simpler and resource-matched baselines explain?

Memory efficiency becomes useful when it changes what evidence can be retained. These controls separate that payoff from ordinary feature selection. The same-K comparisons ask whether bounded leaves preserve a fixed selected set while changing its resource profile. The measured-memory comparison then gives selection-only the widest feasible single leaf under the folded route's observed peak budget. It uses the fixed 18-dataset registry and held-out folds 1–4; widths are selected by memory alone, before prediction metrics are aggregated.

Table 21: **Folding buys capacity with a measured latency cost.** The retained columns, checkpoint, fit, and complete-label prediction budget are identical between a single unbounded leaf and leaf-64 SCFF. Times are medians of three synchronized candidate forwards after one warmup; rows report medians over paired units. Speed is single-leaf time divided by multi-leaf time, and memory is the multi/single ratio of incremental peak CUDA allocation.

Dataset	Units	Retained K	Single ms ↓	Folded ms ↓	Speed ↑	Memory ↓
Internet-Advertisements	10	512	404.2	667.3	0.597×	0.134×
Bioresponse	9	768–832	776.9	950.1	0.781×	0.093×
cnae-9	10	256	127.0	197.3	0.654×	0.259×
gina_agnostic	1	448	302.4	419.0	0.722×	0.149×
All units	30	256–832	401.6	590.3	0.709×	0.134×

All 30 units complete with zero failure. Prediction and retained-feature hashes are checked before aggregation. Folding lowers incremental peak allocation on 30/30 units, to 13.4% of the single-leaf route at the overall median. It is slower on 28/30 units: median same-feature speed is 0.709× (about 1.41× longer). At $K > 512$, the median memory ratio is 0.093×.

Table 22: **Selected evidence under the same measured peak-memory budget.** TabICLv2, one member, $M = 128$, four held-out folds on each dataset in the fixed 18-dataset registry. K is the support-selected Core width; k is the widest single-leaf ANOVA prefix fitting the paired folded peak. Positive accuracy and NLL reduction favor folding.

Dataset	Median K	Median k	ΔAcc. pp	NLL reduction
Arrhythmia	186	64	+0.00	-0.0223
Scene	242	96	+0.00	+0.0003
Sylva	54	54	0.00	0.0000
Gina	391	112	+2.34	+0.0608
Musk	129	80	+3.71	+0.1187
Gas Drift	127	96	-0.02	+0.0007
Gas Drift (concentrations)	129	64	-0.10	-0.0007
Christine	684	96	+0.59	+0.0038
Jasmine	64	64	-0.13	-0.0002
Madeline†	13	13	0.00	0.0000
Philippine	117	117	+0.54	+0.0057
Guillermo	55	55	0.00	0.0000
Riccardo	357	96	+1.54	+0.0358
Dilbert	1,691	128	+3.47	+0.1021
Fabert	229	96	+12.43	+0.2710
Robert	6,753	128	+16.63	+0.4328
Hiva	1	1	0.00	0.0000
KDDCup09 Appetency†	5	5	0.00	0.0000

Dataset-macro differences average the four paired folds. All 18 datasets and 428 width points completed numerically. In 66/72 folds, k satisfies the strict measured-memory inequality. †The other six folds have $K = k \leq 14$ and identical predictions on the two one-leaf routes; single-run peak readings differ by 0.19–1.39%. The strict complete-case estimate retains the 16 datasets with all four folds within budget (64 folds), excluding both marked datasets. The $K > M$ stratum in the main text contains 10 datasets and all 40 of its folds satisfy the measured budget.

The same frozen memory-only prefix grid confirms the capacity comparison on TabPFN-3 under its strict one-member inference contract. All 430 prefix configurations and 72 bounded-Core evaluations completed on the same host; each fold had a feasible single-leaf prefix at or below its bounded peak. The bounded-Core predictions exactly reproduce the prior held-out receipt. Table 23 reports both the complete registry and the width stratum fixed before inspecting outcomes.

Table 23: **TabPFN-3 Core folding at the bounded route's measured peak-memory budget.** The single-leaf control uses the widest support-ranked Core prefix in the fixed grid whose measured peak does not exceed the paired bounded-Core peak. Positive accuracy and NLL reduction favor folding. Intervals resample datasets after averaging the four held-out folds.

Scope	Datasets	Folds	ΔAcc. pp [95% CI]	NLL reduction [95% CI]
All registered wide tables	18	72	$+1.86\ [+0.47, +3.63]$	$0.0431\ [0.0110, 0.0841]$
Core width $K > 128$	9	36	$+3.72\ [+1.29, +6.63]$	$0.0862\ [0.0302, 0.1541]$

The panel, folds 1–4, and width grid $\{16, 32, 64, 96, 128, 192, 256, 384, 512, 768, 1024, 1536, 2048, K\}$ were fixed before aggregation; the grid is clipped to each support-selected K. All 18 datasets and all 72 folds are retained, including 36 same-width identity comparisons. Peak memory is measured on the same eight-card host for both arms; widths are chosen without query labels. Dataset bootstrap uses 100,000 draws and seed 20260920. Bounded-Core candidate-forward time is $2.76\times$ that of the selected narrower prefix on average across datasets.

Table 24: **Memory savings at identical selected width, a different comparison.**

Dataset	K	ΔAcc. pp	ΔNLL	Folded/single peak
Internet-Ads	512	$+0.00$	$+0.0110$	16.3%
Bioresponse	832	$+0.00$	$+0.0081$	10.5%
cnae-9	256	-0.93	-0.0275	35.2%
gina	448	$+0.58$	-0.0137	18.9%

Both arms encode exactly the same K columns with query chunk 128. Deltas are folded minus single leaf. Memory is total peak allocation, not the incremental allocation reported in Table 21. Quality is not uniformly improved at identical K.

Together, these comparisons establish two complementary uses of folding: encode a fixed Core with less memory, or use a fixed memory budget to encode a larger Core. The second use closes the loop from systems efficiency to predictive quality without changing the checkpoint or prediction count. On TabICLv2, removing the two largest-gain datasets still leaves $+0.75$ accuracy points and a 0.0191 NLL reduction across the other 16 datasets. The supplementary numeric records permit reselecting every memory-feasible prefix and recomputing the paired effects.

E Robustness and Resource Analyses

Real wide tables vary in how many columns are informative and how many examples are available. The following interventions change those conditions one at a time: selected Core width, added irrelevant columns, leaf capacity, and support context. The final application asks whether the same resource argument carries over to algorithmically generated features.

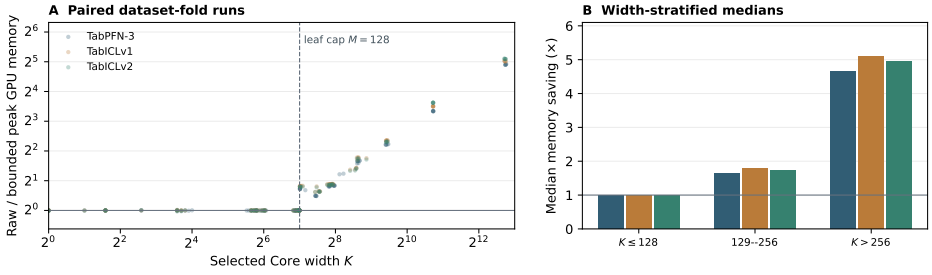


Figure 5: **Core memory savings grow with width.** On 90 paired folds per backbone, median raw/bounded peak-memory ratios for $K > 256$ are $4.66\times$, $5.11\times$, and $4.96\times$ for TabPFN-3, TabICLv1, and TabICLv2; $K \le 128$ is near parity.

E.1 Does folding resist irrelevant width?

More columns do not necessarily mean more evidence. To separate width from information, the nuisance intervention adds split-local, label-independent transformations without deleting them. Doses are nested within seed and the four tasks are fixed before evaluation. Dose zero has one record per task; each of four nonzero doses has three seeds per task, giving 52 records. We average seeds within task and dose and then macro-average tasks. From dose 0 to 512, native accuracy changes by -0.605 points and NLL by $+0.01919$; SCFF changes by -0.085 points and -0.00105 NLL. This intervention supports robustness to the tested nuisance generator and dose range; it does not establish robustness to arbitrary dependent or adversarial columns.

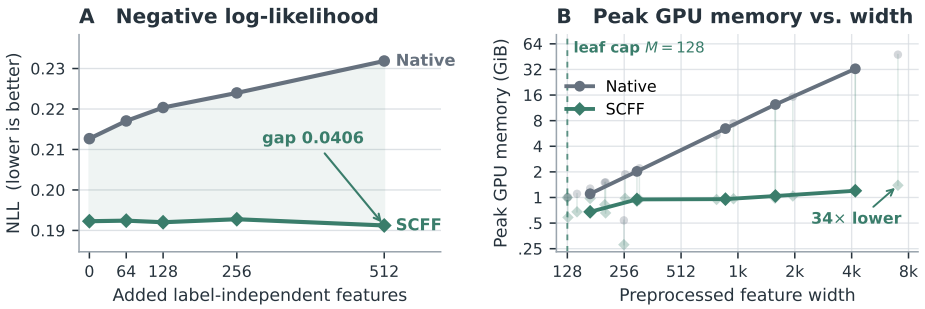


Figure 6: **SCFF resists nuisance width while retaining every column. A:** NLL with 0–512 label-independent columns. **B:** Peak GPU memory as preprocessed width grows. At 512 added columns, SCFF is 1.440 accuracy points and 0.04063 NLL better than native.

E.2 How do leaf width and context interact?

The method needs groups small enough to be affordable but large enough to preserve useful interactions. The leaf ablation changes only $M \in \{32, 64, 128\}$ over four tasks and five folds. The router, Tail update, checkpoint, and splits remain fixed. Main Table 4 shows that $M = 32$ loses quality, while 64 and 128 both improve over native. The pooled point estimate at 64 is slightly better in quality, time, and peak memory; $M = 128$ remains the fold-0-selected default used by the full evaluation. We therefore interpret the sweep as a moderate-width operating range, not evidence that 128 is uniquely optimal.

The context panel asks a different question: whether the effect persists as the number of support rows changes. One dataset is fixed from each raw-width band using input metadata, support prefixes are nested, support order and query rows are fixed, and no quality result enters the context choice. Mixed cells rule out a universal monotone context law.

Table 25: **Context-by-width cells.** Width bands and representatives are fixed from input metadata. Context prefixes reuse the same support order and query rows within each dataset.

Raw-width band	Dataset	Raw D	Context	ΔAcc. pp	ΔNLL	Peak/native	Time/native
128-255	kddcup09_appetency	212	128	+0.000	−0.00791	73.1%	0.70×
128-255	kddcup09_appetency	212	512	+0.000	+0.00408	61.9%	0.70×
128-255	kddcup09_appetency	212	2048	+0.000	−0.00166	54.6%	0.78×
256-511	philippine	308	128	−4.102	+0.08142	57.6%	0.65×
256-511	philippine	308	512	−2.246	+0.00761	45.4%	0.64×
256-511	philippine	308	2048	+0.879	−0.01000	43.0%	0.79×
512-2047	dilbert	2000	128	+5.762	−0.18864	12.9%	1.63×
512-2047	dilbert	2000	512	+2.148	−0.01944	9.7%	1.32×
512-2047	dilbert	2000	2048	−0.098	+0.00049	6.8%	0.79×
2048+	guillermo	4296	128	+2.344	−0.08459	6.8%	2.41×
2048+	guillermo	4296	512	+8.594	−0.10163	4.6%	1.34×
2048+	guillermo	4296	2048	+8.887	−0.11406	3.6%	1.05×

All 12 cells pass exact feature coverage, positive leaf-weight, tail-mass, nested-support, fixed-query, and one-prediction audits. The mixed response of `dilbert` and the large gains on `guillermo` expose context–width heterogeneity rather than a monotone context-scaling law.

E.3 Can saved feature memory provide more context?

Feature width and support length compete for the same device memory. This experiment spends the saving from narrower feature interactions on more labeled examples. The separately registered full-coverage mass-Huber folding study uses an all-feature mass-weighted Huber average, predating the final residual/prototype update. It is a framework-level resource test, not a long-context validation of the final Tail update. For each eligible fold-0 dataset, Native@2,048 defines the measured memory budget. SCFF support sets are nested prefixes, and $C^\star$ is the largest declared prefix within 102% of that budget; quality and latency are hidden until after memory-only selection. The complete $D \geq 2000$, $n \geq 10000$ panel includes four datasets. Three ultra-wide tasks improve substantially,

whereas the 2,000-feature boundary *dilbert* slightly worsens. This contrast reveals the two forces in context reinvestment: more examples can improve inference, but they must compensate for any representation loss introduced by folding.

Table 26: **Complete fixed-memory study with additional context, including the 2,000-feature boundary.**

Dataset	D	Context N / S	Gain	S@2K mem.	S@$C^\star$ mem.	ΔAcc. %	ΔNLL
dilbert	2,000	2,048 / 7,680	3.75×	6.3%	20.4%	−0.195	+0.00406
guillermo	4,296	2,048 / 15,360	7.50×	3.4%	24.4%	+14.453	−0.22615
riccardo	4,296	2,048 / 15,360	7.50×	3.4%	24.0%	+4.785	−0.14345
robert	7,200	2,048 / 7,680	3.75×	3.3%	11.4%	+8.887	−0.26498
Macro mean	–	–	5.62×	4.1%	20.1%	+6.982	−0.15763

All four datasets satisfy the fixed $D \geq 2000$, $n \geq 10000$ eligibility rule. The main text reports the $D \geq 4096$ ultra-wide group; this table includes the complete boundary analysis. All 14 runs completed without fallback or replacement.

E.4 Composing with generated feature spaces

Feature engineering creates another route to wide tables: a modest raw input can expand into many derived columns. Holding the generated table fixed lets us ask whether SCFF helps an ICL model use this larger evidence space more efficiently, separately from the value of feature generation itself.

The TabPrep application panel contains all 11 classification datasets reported as improved by tuned TabPrep in its published RealTabPFN-2.5 comparison. This external opportunity rule is fixed independently of our TabICLv2 outcomes. Each fold-0 variant shares support/query rows: A uses the raw table, B applies native inference to every support-fitted generated feature, and C applies SCFF to that same generated table. C improves NLL on 10 of 11 datasets relative to B while using 24.3% of its peak memory and $0.71\times$ its time. However, the macro C result does not recover the raw-table A baseline. The experiment therefore supports efficient use of a wide generated feature space, not a claim that feature generation is always beneficial. Lower NLL is reported as predictive log-loss improvement, not as a calibration guarantee.

Table 27: **SCFF retains all TabPrep-generated features.** The 11 datasets are fixed by an external TabPrep-positive criterion. A is the raw table, B is native inference on every generated feature, and C applies the selected support-validated Tail update to the same complete generated table. Deltas are dataset-macro right-minus-left changes; W/T/L follows the metric direction.

Comparison	ΔAcc. pp↑	ΔNLL↓	$\Delta\mathrm{AUC}_{\mathrm{OVR}}$ ↑	Acc. W/T/L	NLL W/T/L
B − A: generated features	-0.302	+0.01245	-0.01065	5/1/5	3/0/8
C − A: generated + SCFF	-0.239	+0.00718	-0.00548	4/2/5	5/0/6
C − B: generated features + SCFF	+0.064	-0.00527	+0.00516	4/2/5	10/0/1

Raw widths span 9–170 columns and TabPrep expands them to 136–2,000. Relative to native inference on the generated table, SCFF uses 24.3% of peak GPU memory and 0.71× total prediction time. Every generated feature is encoded exactly once; the largest accepted Tail weight is 1.000000.

Dataset	Raw→generated	Leaves	ΔAcc. pp	ΔNLL	Peak C/B
Amazon_employee_access	9→2,000	16	+0.43	-0.0159	13.7%
APSFailure	170→2,000	17	+0.56	-0.0149	41.5%
bank-marketing	13→677	6	-0.01	-0.0013	20.5%
churn	19→2,000	17	+0.00	-0.0044	7.1%
coil2000_insurance_policies	85→2,000	17	-0.03	-0.0048	7.2%
Diabetes130US	47→2,000	17	-0.02	-0.0017	22.8%
E-CommereShippingData	10→959	8	-0.19	-0.0004	13.8%
polish_companies_bankruptcy	64→2,000	17	+0.05	-0.0062	7.2%
SDSS17	11→2,000	17	+0.15	-0.0085	42.6%
seismic-bumps	15→2,000	17	+0.00	-0.0006	7.6%
NATICUSdroid	86→136	2	-0.24	+0.0006	92.7%

F Where the Design Transfers

The experiments point to three conditions for successful transfer: a compatible representation interface, enough local capacity to encode useful interactions, and Tail evidence that survives the support check. SCFF combines aligned leaf outputs before a shared context encoder. TabPFNv1 is a capacity extension because its standard inference fixes a 100-feature view. LimiX instead uses the token-preserving adapter above. The Tail update improves three of five models, so it does not transfer to every compatible representation interface.

Tail validation currently supports binary tasks; multiclass tasks use Bounded Core. Stored leaf representations require memory that grows linearly with width, the headline locked-fold intervals quantify dataset-level uncertainty for the complete method, and the all-five-fold same-Core intervals are favorable for both metrics on TabICLv2 but cross zero on TabPFN-3 and TabICLv1. In the secondary folds 1–4 contrast, both intervals also favor bounded folding on TabPFN-3, while those on TabICLv1 still cross zero. The smaller incremental Tail effect has favorable intervals for both metrics only on TabICLv2, and the fixed-memory study uses an earlier Tail update.

G Locked-Fold Dataset Effects

The aggregates establish the overall tradeoff; the dataset-level results show how that tradeoff varies across tasks. The following tables report every dataset behind the four matched-width locked-fold intervals, so both gains and regressions can be traced to the same fixed population. Each row is the mean of folds 1–4; no development-fold observation enters these values.

Table 28: **Locked-fold dataset effects for TabPFN-3.** Each row averages folds 1–4; positive Accuracy and negative NLL favor SCFF.

Dataset (OpenML ID)	ΔAccuracy pp	ΔNLL
arrhythmia (5)	-0.571	-0.02489
scene (312)	+0.000	-0.00043
sylva_agnostic (1036)	+0.000	+0.00062
gina_agnostic (1038)	+2.055	-0.04217
musk (1116)	+1.440	-0.12677
gas-drift (1476)	+0.000	+0.00000
gas-drift-different-concentrations (1477)	-0.146	+0.00168
christine (41142)	+0.098	-0.00441
jasmine (41143)	+0.125	+0.00214
madeline (41144)	+0.637	-0.00044
philippine (41145)	-0.757	+0.01648
guillermo (41159)	+4.785	-0.12165
riccardo (41161)	-0.464	+0.02530
dilbert (41163)	+1.099	-0.04067
fabert (41164)	+23.486	-0.53174
robert (41165)	+3.418	-0.07173
hiva_agnostic (46933)	+0.000	-0.00013
kddcup09_appetency (46939)	+0.000	+0.00021

Table 29: **Locked-fold dataset effects for TabICLv1.** Each row averages folds 1–4; positive Accuracy and negative NLL favor SCFF.

Dataset (OpenML ID)	ΔAccuracy pp	ΔNLL
arrhythmia (5)	+4.164	-0.08222
scene (312)	-1.507	+0.02088
sylva_agnostic (1036)	+0.488	-0.00941
gina_agnostic (1038)	+9.264	-0.19215
musk (1116)	+0.659	-0.00627
gas-drift (1476)	-0.024	+0.00120
gas-drift-different-concentrations (1477)	+0.000	-0.00301
christine (41142)	+3.320	-0.03435
jasmine (41143)	+2.011	-0.01301
madeline (41144)	+31.131	-0.40828
philippine (41145)	+5.591	-0.08871
guillermo (41159)	+18.945	-0.20223
riccardo (41161)	-2.637	+0.01651
dilbert (41163)	+5.737	-0.13595
fabert (41164)	+20.605	-0.42780
robert (41165)	-1.465	-0.02031
hiva_agnostic (46933)	+0.000	-0.00039
kddcup09_appetency (46939)	-0.024	-0.00361

Table 30: **Locked-fold dataset effects for TabICLv2.** Each row averages folds 1–4; positive Accuracy and negative NLL favor SCFF.

Dataset (OpenML ID)	ΔAccuracy pp	ΔNLL
arrhythmia (5)	+2.485	-0.02400
scene (312)	+1.143	-0.02215
sylva_agnostic (1036)	+0.000	-0.00023
gina_agnostic (1038)	+0.865	-0.01761
musk (1116)	+2.222	-0.09004
gas-drift (1476)	+0.000	-0.00012
gas-drift-different-concentrations (1477)	-0.098	+0.00224
christine (41142)	+0.610	-0.01382
jasmine (41143)	+0.125	-0.00066
madeline (41144)	+0.119	+0.00197
philippine (41145)	+1.123	-0.01562
guillermo (41159)	+9.814	-0.12496
riccardo (41161)	+0.684	-0.03448
dilbert (41163)	+0.391	-0.01129
fabert (41164)	+1.489	-0.07060
robert (41165)	+8.032	-0.19126
hiva_agnostic (46933)	+0.065	-0.03949
kddcup09_appetency (46939)	+0.000	+0.00351

Table 31: **Locked-fold dataset effects for RefineICL-24L.** Each row averages folds 1–4; positive Accuracy and negative NLL favor SCFF.

Dataset (OpenML ID)	ΔAccuracy pp	ΔNLL
arrhythmia (5)	+6.364	-0.21560
scene (312)	-0.000	+0.00352
sylva_agnostic (1036)	-0.049	+0.00094
gina_agnostic (1038)	+0.036	-0.00611
musk (1116)	+3.491	-0.09720
gas-drift (1476)	-0.049	+0.00226
gas-drift-different-concentrations (1477)	-0.024	-0.00120
christine (41142)	+4.077	-0.13376
jasmine (41143)	-0.545	+0.01392
madeline (41144)	-4.498	+0.08619
philippine (41145)	-1.099	+0.01959
guillermo (41159)	+10.938	-0.15733
riccardo (41161)	+42.114	-0.69906
dilbert (41163)	+1.758	-0.04319
fabert (41164)	-1.733	+0.01033
robert (41165)	+8.447	-0.29305
hiva_agnostic (46933)	+0.000	-0.00311
kddcup09_appetency (46939)	+0.000	+0.00198

Table 32: **Typical locked-fold dataset effects.** Each entry first averages folds 1–4 within a dataset. Medians and win/tie/loss counts describe datasets, not individual query rows; positive Accuracy and negative NLL favor SCFF.

Backbone	Median ΔAcc. pp	Acc. W/T/L	Median ΔNLL	NLL W/T/L
TabPFN-3	+0.049	9/5/4	-0.00044	11/1/6
TabICLv1	+1.335	11/2/5	-0.01666	15/0/3
TabICLv2	+0.647	14/3/1	-0.01662	15/0/3
RefineICL-24L	0.000	8/3/7	-0.00216	10/0/8

H REPRODUCIBILITY

The reproduction materials follow the paper's comparisons: complete-method quality, fixed-Core folding, incremental Tail recovery, and resource reinvestment. Each table can be traced to its paired records and regenerated under the stated inference contract.

The final residual/prototype evaluation is identified by `scff_core_raw_tail_signal_optimization_v1`; the separately labeled fixed-memory experiment uses the all-feature mass-weighted Huber average described above. Reproduction requires the checkpoint hashes, OpenML IDs, fold records, support/query hashes, preprocessing records, method and adapter revisions, and runtime fingerprints. Paired fold-level metrics, failure histories, and table-generation scripts are released without dataset rows or per-query predictions. The appendix reports results from the final method and the separately labeled fixed-memory study.

Within the supplementary archive, the portable locked-fold metric root is `paper_evidence/artifacts/scff_paper/core_raw_tail_signal_v1/`; `full_method_all_five_folds.json` indexes the paired fold-level metrics and `tail_confirmation_folds1_4.json` summarizes the locked Tail comparison. `core_confirmation_bootstrap_receipt.json` records the same-Core dataset-bootstrap seed, sample count, estimands, and intervals. The scripts `render_scff_heldout_full_method_table.py` and `render_scff_influence_sensitivity.py` verify coverage and source hashes before regenerating the reported tables. The RefineICL-24L row has a paired metric record, checkpoint digest, and separate anonymous training-source archive here. Its exact concurrent checkpoint is not redistributed; independent re-execution of that backbone awaits the companion weights.