TABULAR FOUNDATION MODELS ARE EFFECTIVELY SHALLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context-learning Tabular Foundation Models (TFMs) are uniquely suited to scenarios with limited resources or privacy constraints, as they solve new tasks without additional training. However, their architecture consists of multiple transformer blocks, resulting in high computational and memory footprints. Meanwhile, traditional alternatives like Gradient-Boosted Decision Trees (GBDTs) remain highly competitive and run efficiently on CPUs, suggesting that tabular tasks may not inherently require deep architectures. This contrast raises a fundamental question: *can TFMs be simplified by removing architectural redundancies without sacrificing performance?* To answer this, we introduce a closed-form, label-free affine translator that replaces a contiguous window of up to 96% of the blocks, leaving the rest unchanged. Evaluating eight frontier TFMs on 171 classification datasets, we reveal that tabular transformers are vastly overparameterized: replacing half of a model's blocks costs at most 0.002 median AUROC across all architectures, with the half-depth TabFM still outperforming every other full-depth TFM and tuned GBDTs. Furthermore, different model families are overparameterized in different ways: TabPFN-3 relies on its encoders, maintaining performance regardless of which single block is retained. Other single-axis models spread computation thinly, needing only their first block, while dual-axis models concentrate it in one critical block. Ultimately, removing depth saves up to 1.5B parameters, speeds up GPU inference by up to $6\times$ at a median cost of 0.03 AUROC, while a single-block TabPFN-3 serves cached queries $2.4\times$ faster using $6\times$ less memory at negligible cost.

1 INTRODUCTION

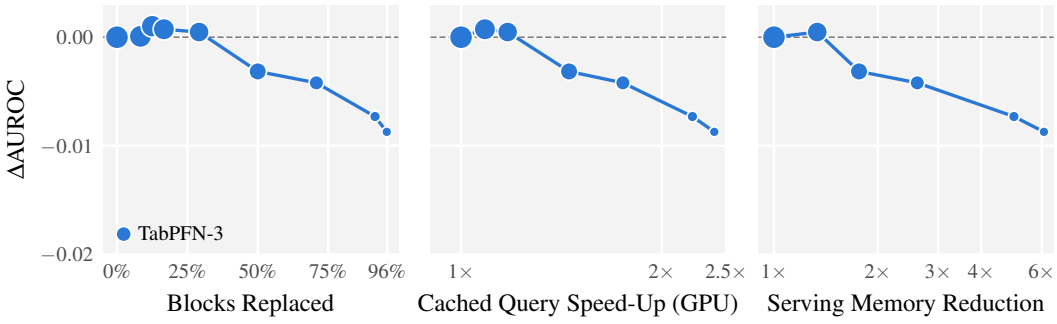

Figure 1: **TabPFN-3 at different levels of compression.** Change in the classification AUROC using MIMIC-III vs. (left) % of replaced blocks, (middle) cached-query GPU speed-up, and (right) serving-memory reduction. Keeping only one transformer block (96% of blocks replaced) costs -0.009 AUROC for a $2.4\times$ speed-up and $6.1\times$ less memory.

In-context-learning (ICL) TFMs offer a powerful approach for tabular prediction: they solve new tasks with no additional training, handle heterogeneous data, and require no preprocessing or hyperparameter tuning. Recent models such as TabPFN (Hollmann et al., 2025; Grinsztajn et al., 2026), TabICL (Qu et al., 2025), and TabDPT (Ma et al., 2025) achieve strong performance across diverse

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.

benchmarks. However, this zero-shot convenience comes at a high computational cost. Current TFMs stack 12 to 24 identical transformer blocks on top of their encoders. This depth imposes heavy memory footprints and high inference latency, limiting deployment on resource-constrained hardware such as local clinical servers. Meanwhile, traditional alternatives like GBDTs remain highly competitive on these same tasks while running efficiently on CPUs (Grinsztajn et al., 2022; McElfresh et al., 2023). The success of these shallower, tree-based models suggests that typical tabular tasks do not inherently require deep hierarchical composition. If that is true, much of the deep architecture inside current TFMs may be computationally redundant. This contrast raises a fundamental question: *can we retain the zero-shot convenience of TFMs while approaching the efficiency of simpler models by removing architectural redundancy?*

To answer this, we must first understand how these models utilize their depth. While observational metrics such as pairwise representation similarity and attention profiling reveal signs of architectural redundancy, the correspondence between these signals and actual replaceability is inconsistent across models. We therefore locate redundancy by intervening on the architecture itself: we introduce a closed-form, label-free affine translator that bypasses a contiguous window of transformer blocks. If the output distribution remains stable, the skipped blocks were performing computations that a simple linear map can absorb; if the output collapses, they were performing necessary nonlinear composition. We design this translator with a per-feature parameterization to support dual-axis attention, and select the optimal bypass window by minimizing the forward KL divergence on a small calibration split.

Evaluating 8 TFMs from 6 model families across 171 classification datasets, we show that current tabular architectures are overparameterized. Replacing half the blocks across all models results in a minimal median drop of at most 0.002 AUROC, with up to 97% of datasets staying within one point of the full model's performance. Pushing compression to a single block maintains high performance for TabPFN-3, Orion-Bix, and TabICL, which stay within a median 0.007 AUROC of their full-depth counterparts. The savings translate directly to hardware efficiency: halving the depth doubles the throughput of the transformer stack, and with a cached context, per-query GPU latency drops by $1.5\times$–$1.9\times$ at Half and $2.4\times$–$5.1\times$ with one block kept. We show that naively dropping blocks causes model collapse, and early exit performs worse than the label-free translator in most configurations. The translator can also be reused across similar datasets. For example, a half-depth TabICL translator fitted on MIMIC-III (Johnson et al., 2016) transfers to the eICU (Pollard et al., 2018) dataset with an AUROC penalty of just 0.004 (against 0.003 for a local fit), though transferability diminishes between datasets with different schemas (e.g., Heart and Diabetes, Section B.14).

Finally, through a controlled pre-training study on nanoTabPFN (Pfefferle et al., 2025), we investigate whether compression matches natively shallow training. We observe that a 12-layer model compressed to k layers matches the performance of a model trained from scratch at depth k (within 0.011 AUROC). Furthermore, a natively trained 6-layer model matches the AUROC of the full 12-layer baseline, indicating that depth is over-provisioned in this specific setting. Whether this structural redundancy scales to significantly deeper architectures remains an open question for future work. For existing models, our proposed approach offers a practical method to achieve the inference efficiency of a shallow architecture without the computational cost of retraining from scratch.

2 Related work

Tabular Transformers. The development of tabular transformers began with task-specific architectures. Early adaptations like FT-Transformer (Gorishniy et al., 2021) and SAINT (Somepalli et al., 2021) apply standard attention mechanisms to supervised learning, while TabNet (Arik & Pfister, 2021) uses sequential sparse routing for step-wise feature selection. Since then, the field has moved toward general-purpose foundation models, with a strong focus on in-context learning TFMs. TabPFNv2 (Hollmann et al., 2025) and TabPFN-3 (Grinsztajn et al., 2026) scale prior-data fitted networks to handle heterogeneous columns and mixed data types. TabICL (Qu et al., 2025; 2026) processes tabular structures via set-transformers. TabDPT (Ma et al., 2025) pre-trains on real data, while TabFM (Google Research, 2026) is pre-trained on synthetic data, and Mitra (Zhang et al., 2025b) mixes diverse synthetic priors. Finally, the Orion family (Bouadi et al., 2025; 2026) and LimiX (Zhang et al., 2025a) vary the attention pattern of the same in-context recipe (multi-scale sparse, bi-axial and dual-axis cell attention).

Transformer Redundancy. Over-parameterization is a well-documented trait of vision and language transformers, where dropping layers, attention heads, or tokens often yields minimal performance degradation (Gromov et al., 2025; Men et al., 2025; Michel et al., 2019; Bolya et al., 2023; Venkataramanan et al., 2024). In vision, Jacobs et al. (2026) replace deep transformers with a small number of recurrent blocks, while Bolya et al. (2023) explicitly exploit token-level redundancy. More closely related to our approach, Shopkhoev et al. (2025) and Cannistraci et al. (2026) show that training-free linear maps can accurately bridge skipped block representations. Following the recent emergence of ICL TFMs, several concurrent works have begun exploring depth and context redundancy in the tabular domain. However, concurrent approaches fundamentally differ from ours by requiring either additional pre-training or dataset-specific gradient updates. For instance, Balef et al. (2026) pre-train a single-layer model from scratch to loop at inference, while Zabërgja et al. (2026) pre-train a predictor-compressor pair to condense context rows into latent vectors. Among post-training approaches, TACTICL (Koshil et al., 2026) replaces transformer layers with lightweight MLP adapters trained via gradient descent on the downstream task. Similarly, (Tanna et al., 2026) relies on labeled data to train task-specific student models, while early-exit methods (Küken et al., 2025) pre-train new layer-wise decoders on prior data. In contrast, our approach uses a single, closed-form affine map, requiring no non-linear adapters, gradient updates, or labeled data

3 METHOD

Let $\mathbf{H}_l \in \mathbb{R}^{N\times d}$ denote the latent representations at depth $l \in \{0,\dots,L\}$, where $\mathbf{H}_0$ represents the output of the input embedding layer and $\mathbf{H}_L$ is the output of the final transformer block. For a contiguous window of blocks $0 \le s < e \le L$, we bypass blocks s through $e-1$ entirely. We replace this bypassed window with an affine map $(\mathbf{W}^*, \mathbf{b}^*)$, fitted via ridge regression on an unlabeled calibration set:

$$(\mathbf{W}^*, \mathbf{b}^*) = \arg\min_{\mathbf{W},\mathbf{b}} \|\mathbf{H}_e - (\mathbf{H}_s\mathbf{W} + \mathbf{b})\|_F^2 + \lambda \|\mathbf{W}\|_F^2. \tag{1}$$

The hidden states $\mathbf{H}_s$ and $\mathbf{H}_e$ are extracted using a single forward pass over the in-context support set and the unlabeled calibration queries. To bound the solve time independently of context size, we fit the map on a fixed random subset of $N = \min(500, n_{\text{support}} + n_{\text{cal}})$ of these sequence positions. The regularization parameter $\lambda \in \{0, 10^{-3}, 10^{-2}, 10^{-1}\}$ is scaled relative to the mean diagonal of the Gram matrix. We evaluate the fitting residual on a 25% held-out validation split of these calibration rows to select λ^*, then refit the map on the full subset.

At inference, this closed-form map acts as a linear module projecting $\mathbf{H}_s$ directly to $\mathbf{H}_e$, while the remaining blocks $e,\dots,L-1$ and the original classification head operate entirely unchanged. The closed-form fit adds negligible overhead to context encoding. Additionally, refining the map by gradient descent does not improve it (Section A.3).

Per-Architecture Parameterization. Single-axis self-attention architectures such as TabFM and TabPFN-3 operate on sequence tensors where items are processed independently, requiring a single shared transformation matrix $(\mathbf{W}, \mathbf{b})$. Conversely, dual-axis models such as TabPFNv2 and LimiX attend across both rows and features, producing hidden tensors of shape $(\cdot, n_{\text{features}}, d)$. Applying a single shared matrix across this tensor would mix feature representations and disrupt axis alignment. Thus, for dual-axis architectures, we fit a distinct affine map $(\mathbf{W}_f, \mathbf{b}_f)$ for each feature index $f \in \{1,\dots,n_{\text{features}}\}$ (additional details in Section A.3). This expands the total translator parameterization to $n_{\text{features}}(d^2+d)$, which scales dynamically with the dataset's feature dimensionality. However, in some cases (e.g., TabPFNv2), these parameters can be largely absorbed.

Absorbing the Map. Because the map is affine, it can be absorbed into an adjacent affine layer. However, since transformer blocks contain non-linear operations, this operation is restricted to the network boundaries. If the window ends at the final block and the classification head applies a linear projection $\mathbf{x} \mapsto \mathbf{x}\mathbf{A}^\top + \mathbf{c}$, the map folds exactly as $\mathbf{H}_s(\mathbf{W}^*\mathbf{A}^\top) + (\mathbf{b}^*\mathbf{A}^\top + \mathbf{c})$. Symmetrically, if the window starts at block 0, the map folds into the encoder's final linear layer. In these cases, the compressed model strictly reduces to a standard k-block architecture with no additional parameters. Additional details are provided in Section B.16.

Label-Free Window Selection. To determine which block window to bypass, we evaluate candidate window bounds (s, e) on the calibration split X_{cal}. Because our bypass retains the original pre-trained classification head, candidate models produce directly comparable probability distributions. Inspired by the KL-guided layer selection of Li et al. (2026), for a target width k, we select the optimal window (s^*, e^*) by minimizing the forward KL divergence between the uncompressed model's output distribution p_{full} and the compressed model's distribution $p_{(s,e)}$:

$$(s^*, e^*) = \arg \min_{(s,e):\, e-s=k} \mathbb{E}_{x \in X_{\text{cal}}} \left[\mathrm{KL} \left(p_{\text{full}}(y \mid x) \,||\, p_{(s,e)}(y \mid x) \right) \right]. \tag{2}$$

Minimizing output KL divergence rather than the latent residual (e.g., $\|\mathbf{H}_e - \hat{\mathbf{H}}_e\|_F^2$) is advantageous for two reasons. First, latent distances weigh all error directions equally, whereas the KL divergence weighs each error by its effect on the final predictions, after the remaining blocks and the classification head have amplified or suppressed it. Second, hidden-state scales differ across depths, so latent residuals of windows ending at different blocks are not comparable; the KL divergence scores every candidate in the same output probability space, across window positions and widths.

Window selection is executed as a one-time pre-computation during context encoding. Candidate windows share a single hooked forward pass over X_{cal} to cache intermediate states. For each candidate (s, e), fitting the affine map and scoring the resulting distribution requires only a closed-form matrix solve and a fast pass through the remaining $L - e$ layers. Once selected, the affine map is frozen for all subsequent queries. Single-axis models permit a static default selection, whereas dual-axis ones exhibit dataset-specific layer sensitivities that benefit from dynamic selection (Table 7). The same criterion can also choose how many blocks to replace. Given a divergence tolerance $\epsilon > 0$, we select the widest window whose calibration drift stays within ϵ:

$$\max_{s,e} (e - s) \quad \text{s.t.} \quad \mathbb{E}_{x \in X_{\text{cal}}} \left[\mathrm{KL} \left(p_{\text{full}} \,||\, p_{(s,e)} \right) \right] \leq \epsilon. \tag{3}$$

At $\epsilon = 0.01$, this rule removes 52–70% of the blocks of the single-axis models at a mean cost of at most 0.004 AUROC (Section B.4).

4 Locating Computational Redundancy

Before evaluating the performance of compressed models, we investigate how computation is distributed across the transformer blocks. We first examine representation and attention dynamics, then we perform a per-block intervention to accurately locate redundant blocks to replace.

Experimental Setup. We evaluate 8 pre-trained TFMs from 6 model families (TabPFN, TabICL, TabDPT, TabFM, Orion and LimiX) with frozen weights across 171 classification datasets: 163 from four deduplicated benchmark suites (OpenML-CC18 (Bischl et al., 2021), TabZilla (McElfresh et al., 2023), TabArena (Erickson et al., 2025), Grinsztajn (Grinsztajn et al., 2022)) and eight clinical datasets including MIMIC-III (Johnson et al., 2016) and eICU-CRD (Pollard et al., 2018). Each dataset is split into train, calibration, and test, and run at 3 seeds. For deterministic extraction, where possible, prediction-time ensembling is reduced to a single member, and LimiX-16M's input preprocessing is fitted once on the support rows and then frozen (Section A.3). Additional details, including datasets and models details, are in Section A, while additional results are in Section B.

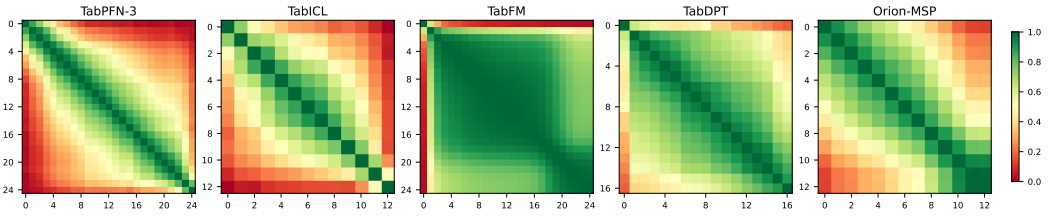

Figure 2: **Pairwise Cosine Similarity.** Layerwise cosine similarity between all block representations using MIMIC-III across different TFMs. Except for TabFM, models show architecture-specific structure that does not reliably predict which blocks are replaceable.

Representation Similarity. Figure 2 shows the full pairwise cosine similarity between every pair of block representations on a single forward pass using MIMIC-III. The resulting matrices vary drastically across architectures. For example, from block 3 on, consecutive TabFM representations have cosine similarity ≥ 0.99, consistent with the concurrent findings of Balef et al. (2026) that other TFMs representations stabilize early. In contrast, TabDPT shows broad similarity in its middle blocks and Orion-MSP in its last ones, while TabPFN-3 exhibits a smooth transition with no sharp boundaries. Finally, TabICL's final block is the most dissimilar from every other block, while results in Figure 11 show that it is fully replaceable by the affine translator. Representation similarity therefore does not accurately predict replaceability.

Attention Patterns. If the depth of a tabular TFM is redundant, one might expect later blocks to simply stop specializing and repeatedly retrieve the same context rows. However, attention statistics on two diverse datasets, MIMIC-III and Higgs, show a different behaviour: the five most-attended keys per query overlap by a Jaccard index of only 0.10–0.19 between consecutive blocks (vs. 0.005 randomly). Additionally, TabPFN-3's first block shares almost no keys with later ones (Section B.1). Nor does attention specialize with depth as in language models. TFMs attend almost uniformly over their context at every block (averaging 81–95% of maximum entropy, and $\geq$70% at every block except TabFM's read-out block), whereas Qwen3-4B's attention is sharply peaked and highly variable (Figure 3 and Section B.1). Ultimately, although each block retrieves a distinct set of rows, the entropy profile remains persistently flat. Consequently, we must look beyond attention mechanisms or embedding similarities to understand when the network becomes redundant.

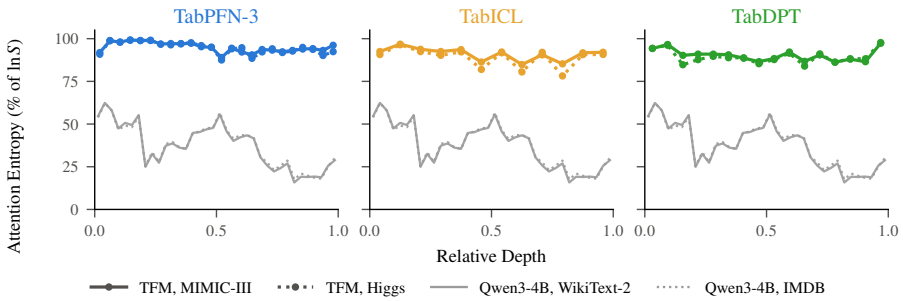

Figure 3: **Per-block Attention Entropy.** Normalized attention entropy (percentage of maximum $\ln S$) for three representative TFMs (on MIMIC-III and Higgs) vs. Qwen3-4B. While tabular models attend near-uniformly to all tokens, Qwen3-4B displays more peaked attention.

Per-Block Sensitivity. To locate the redundant blocks directly, we use the affine translator itself as diagnostic tool. Specifically, we intervene by replacing individual blocks with the translator and measuring the resulting change in AUROC (Figure 4). Notably, the surviving blocks behave just as they do in the full model (attention entropy shifts by $\leq 2\%$; Section B.1). For single-axis models, no single block is critical and every replacement stays within 0.007 AUROC of the full model. In contrast, dual-axis models each have exactly one critical block (i.e., block 0 for TabPFNv2 and block 3 for LimiX-16M), while every other block stays within 0.043 and 0.009, respectively. In short, evaluated one block at a time, almost the entire stack is affine-replaceable and the non-affine computation that remains sits in a single block of the dual-axis models. More results in Figure 11.

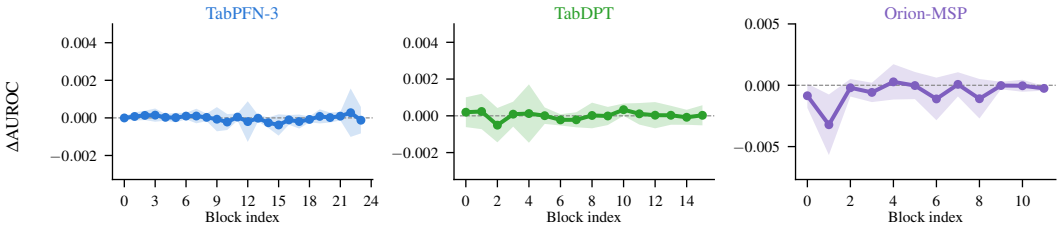

Figure 4: **Per-block Sensitivity.** ΔAUROC when a single block is replaced by the translator. Mean across 8 clinical datasets and 3 seeds. Full results in Figure 11.

5 Evaluation of Compressed Models

Building on Section 4, we now evaluate the predictive performance of our compressed models. Although we analyze the impact of replacing contiguous block windows of varying widths ranging from 0% to 96% of the network (Figures 1 and 12), for clarity we focus our discussion on two primary configurations: (i) *Half*, a safe, deployable setting that replaces 50% of the blocks with the affine translator, and (ii) *One-block*, an extreme setting that retains only a single block alongside the translator. Additional results are in Tables 9 and 10.

5.1 Classification

Figure 5 shows the change in AUROC (compressed − full) on each of the 163 suite datasets, averaged over three seeds, for every model at Half, at One-block, and at One-block with the identity in place of the translator. In the left panel (*Half*), replacing 50% of the transformer stack with the single affine translator causes practically no performance loss. Across all architectures and dataset scales, the mean AUROC drops by a negligible 0.001 to 0.005, with median losses of at most 0.002. The stability holds on the larger datasets too, and the ranking of the models, among themselves and against tuned GBDTs, is unchanged (Section B.7). Discrete metrics reflect the same stability: accuracy and balanced accuracy drop by at most 0.010 and 0.018 across all tasks (Section B.9).

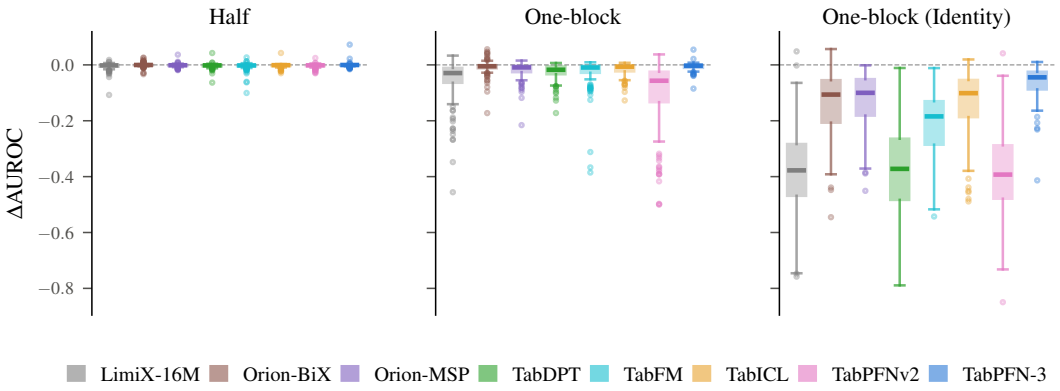

Figure 5: **Classification Performance at Varying Compression Levels.** Per-dataset ΔAUROC (compressed − full) over 163 datasets at half-depth (left), reduced to a single block using our translator (center), and a single-block baseline using an identity mapping (right). Additional results in Tables 9 and 10.

With a single block kept (*center*), the models separate into three groups. TabPFN-3 and Orion-Bix remain nearly lossless, while TabICL, Orion-MSP, TabFM, and TabDPT suffer negligible mean losses (0.015 to 0.026) with some visible outliers, and dual-axis models degrade more, particularly on large datasets and long contexts (Section B.13). TabPFNv2 suffers the most (mean drop 0.101), and LimiX-16M degrades more mildly (mean drop 0.052).

Additionally, this extreme compression isolates exactly where different architectures perform their core computations (Table 17). Single-axis models strongly prefer retaining their first block, relying on it for the initial non-linear attention pass over the context rows. Conversely, dual-axis models must retain their last block to ensure the final representations match the exact distribution expected by the classification head. With a single contiguous window, the kept block is either the first or the last. For TabPFNv2, keeping its critical block 0 costs 0.281 AUROC, while replacing it and keeping the last block costs only 0.071 AUROC. LimiX-16M's critical block 3 is neither first nor last. The map must therefore absorb the critical block's computation, which is why the dual-axis models lose the most at *One-block*; keeping the critical block between two maps would require two windows, which we leave as feature work. TabPFN-3 remains indifferent, as its core computation is already localized in its encoders. Finally, the failures of *One-block* indicate when depth is needed. The cost grows with dataset size (Spearman $\rho = -0.22$; Section B.13), and the largest loss averaged over the six single-axis models is on `monks-2` (-0.128), whose label is a logical rule over its features.

Ablations. One could argue that the skipped blocks are simply inactive, making our translator unnecessary. To test this, we evaluate dropping the same blocks without applying the affine map, effectively treating the skipped section as an identity function (Figure 5, *right*). Under this baseline, performance collapses for most models down to chance level (Table 9). The skipped blocks are therefore not idle, but the closed-form map recovers their computation.

Because an affine approximation suffices, we also evaluate a standard early-exit baseline: fitting a supervised logistic regression classifier directly to the representations right before the skipped window, thereby discarding the remaining blocks and the original head. Despite having access to the support labels, this shortcut underperforms in most configurations. It loses ≈ 0.02 AUROC at every depth, whereas our label-free translator matches or outperforms it in 81% of the 2,664 tested configurations (Table 16). Finally, repeatedly looping the retained block also fails to substitute the affine map (Table 17). Mapping intermediate representations into the space the remaining blocks and the frozen head expect therefore beats training a new head from scratch, even though the new head sees labels.

5.2 Regression

Since Section 5.1 shows that up to 96% of a tabular transformer's depth can be safely removed, we now investigate whether this extreme redundancy is a general property of tabular models or specific to classification. We evaluate the regressor variants of our architectures across 41 continuous datasets. For these experiments, we select the skip window using calibration R^2, as regressors lack the output distributions necessary for the KL divergence criterion. Additional results in Table 20.

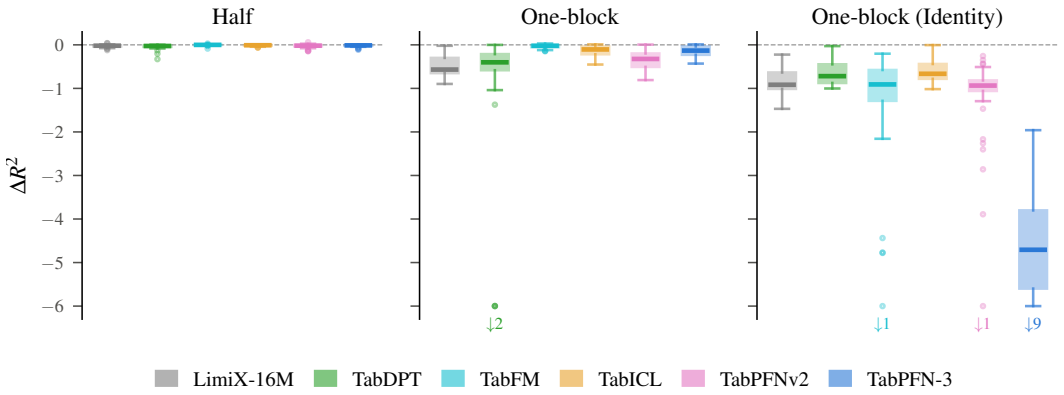

Figure 6: **Regression Performance at Varying Compression Levels.** Per-dataset ΔR^2 (compressed − full) over 41 regression datasets at half-depth (left), reduced to a single block using our translator (center), and a single-block baseline using an identity mapping (right).

Regression shows the same redundancy at *Half* (Figure 6): replacing half of the blocks costs only 0.001 to 0.026 in median R^2, whereas dropping them without the map costs far more (*right*). With one block kept, however, most regressors degrade sharply (e.g., TabPFN-3 loses 0.132 R^2), with TabFM the exception (-0.022 despite going from 24 blocks to 1). Two effects contribute. The predictions shrink toward the mean (standard deviation 0.48–0.72 of the full model's; 0.22 for LimiX-16M), which an affine recalibration on the calibration split partly repairs (Table 21). The samples are also ranked less accurately (Spearman ρ from 0.93 to 0.86 for TabICL and from 0.91 to 0.57 for TabDPT). The late blocks of a regressor therefore refine the ordering of its outputs, which regression needs but classification does not, as long as the decision boundary is preserved. This is why classifiers survive a single block while most regressors need about half of their depth.

6 Efficiency

Finally, we analyze how this compression affects computational cost. Table 1 summarizes the hardware impact of compression on a single RTX 5090. Because the transformer stack accounts for 93%–98% of total weights in most TFMs (except Orion-MSP, at 57%), dropping blocks removes

most of the weights. For single-axis models, the affine translator adds negligible overhead (e.g., 262k parameters replacing 23.6M in TabICL), allowing *One-block* compression to eliminate 91%–96% of stack weights. At this extreme depth, TabFM sheds roughly 1.5B parameters. Dual-axis models present an exception: because their affine map scales with the number of feature tokens, the translator overhead grows on wider datasets, moderating the net parameter reductions (yielding 70%–71% stack savings at *One-block* on MIMIC-III). This map overhead can often be absorbed by adjacent components (Section 3). For example, TabPFNv2 folds its *One-block* map exactly into the encoder, shrinking it from 1.45M to 43k parameters (boosting stack savings from 71% to 91%), while TabDPT folds its 0.59M-parameter map into the classification head. Other models require normalization at these boundaries, retaining their already small maps. For LimiX-16M, whose encoder ends in a LayerNorm, the map cannot be absorbed and grows by 37k parameters per feature token, so on very wide datasets *One-block* LimiX-16M measures redundancy rather than saving parameters.

Table 1: **Hardware Impact.** MIMIC-III, single RTX 5090. Blocks params: parameters of the transformer blocks kept, translator included. ΔFLOPs: forward-pass FLOPs removed. Re-encode: batch of 512 queries, context re-encoded on every call. Cached: one query against a cached context (three models); memory = weights + cache.

	Blocks params (M)			ΔFLOPs		Re-encode speed-up		Cached query speed-up		Cached memory reduction	
Model	Full	Half	One-block	Half	One-block	Half	One-block	Half	One-block	Half	One-block
TabPFNv2	7.1	5.0	2.0	−50%	−91%	1.7×	4.6×	1.9×	5.1×	1.9×	10.0×
TabPFN-3	51.4	25.9	2.4	−26%	−51%	1.1×	1.2×	1.5×	2.4×	1.8×	6.1×
TabICL	25.7	13.1	2.4	−23%	−42%	1.0×	1.1×	1.7×	4.0×	1.7×	4.3×
TabDPT	75.6	38.4	5.3	−49%	−92%	1.5×	3.3×	—	—	—	—
TabFM	1,611	810	71.3	−31%	−59%	1.2×	1.4×	—	—	—	—
Orion-MSP	25.2	12.9	2.4	−22%	−37%	1.1×	1.2×	—	—	—	—
Orion-Bix	25.2	12.9	2.4	−21%	−40%	1.2×	1.3×	—	—	—	—
LimiX-16M	15.9	11.4	4.8	−50%	−91%	1.8×	6.1×	—	—	—	—

While the compressed blocks run up to 23× faster (Table 24), the end-to-end speed-up is limited by the time spent outside them, mostly in the encoders (Amdahl's law). At *One-block*, models with lightweight encoders (TabPFNv2, TabDPT, LimiX-16M) accelerate by 3.3×–6.1×, whereas heavy encoders (TabICL, TabFM) limit gains to 1.1×–1.4×; for these models the main gain is memory, e.g. *One-block* TabFM drops about 6 GB of fp32 weights. Context caching removes this bottleneck, yielding query speed-ups of up to 5.1×. On CPUs, where the blocks dominate the run time, end-to-end speed-ups reach 12.4× (Table 23). The same gains apply to regression, which uses the same backbones: at *Half*, where regression is near-lossless, the lightweight models accelerate by 1.5×–1.8× on GPU, and at *One-block* TabFM sheds 1.5B parameters for a median ΔR^2 of only -0.022.

7 Compression vs. Pre-Training Shallow Architectures

If up to 96% of a tabular transformer's layers can be safely approximated, a natural follow up question would be: why not simply pre-train a shallow model from scratch? To analyze this, we test whether compressing a model matches the performance of a natively shallow one using the nanoTabPFN reference implementation (Pfefferle et al., 2025)[1]. We train the model across varying depths $L \in \{1, 2, 3, 6, 12\}$, each with an embedding dimension of 96 and 4 attention heads, for 5,000 steps across three random seeds (see Section B.17 for configurations). We evaluate these models on four binary clinical datasets (Diabetes, Heart, Breast Cancer, SUPPORT2), strictly operating in the prior-fitting regime (100 support rows, top 5 F-score features, 1,000 test rows). We compare these natively shallow models against the 12-layer model compressed to k blocks using our per-feature translator. Although scaled down (1.4M parameters), this proxy model provides a controlled environment to study the architectural origins of redundancy.

Within this controlled experimental setting, results in Figure 7 suggest three properties of TFMs. First, compressing a pre-trained model performs on par with pre-training a shallow one: AUROC matches within 0.011 at every depth, including a single block. Second, this parity does not come from the translator's extra parameters: training the compressed architecture from scratch, i.e., k blocks plus an identity-initialized $d \times d$ layer ("trained, compressed topology"), does not reliably

[1] `github.com/automl/nanoTabPFN`, Apache-2.0; utilizing a publicly available dataset of 300k synthetic binary tasks (150×5).

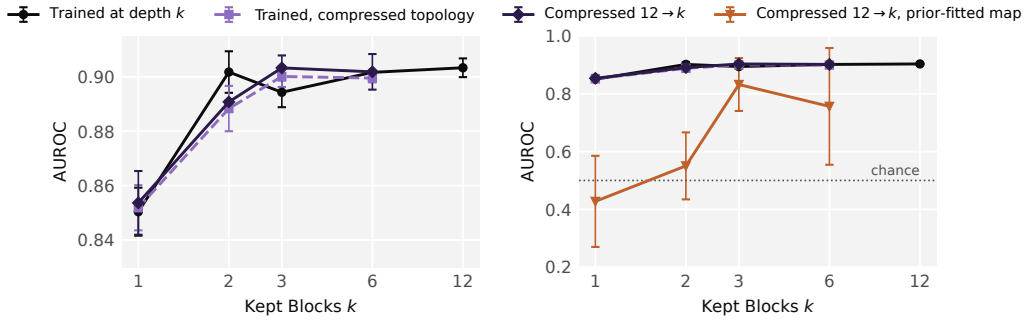

Figure 7: **Pre-training Redundancies.** nanoTabPFN test AUROC, averaged over four clinical datasets. Compressing 12 blocks to k matches native training at depth k, but only when the translator is fitted on the target's (unlabelled) rows rather than on prior tasks (right). Exact values in Table 25.

beat the plain k-block model (0.888 vs. 0.902 at $k = 2$; 0.900 vs. 0.894 at $k = 3$). It does, however, require fitting the map to the target: fitting the map on 20 unseen prior tasks instead of the target's rows drops AUROC to 0.43–0.83 (Figure 7, right), consistent with the transfer study (Section B.14). Third, depth is over-provisioned: a natively trained 6-block model matches the 12-block one, and deeper models are harder to train, collapsing at the default learning rate (Section B.17). With 1,000 support rows, about $7\times$ the pre-training task size, extra depth still does not help, although shallow models trained from scratch lead compressed ones by 0.016–0.027 at $k = 2$ (Section B.18).

8 Conclusion

We introduced a closed-form, label-free affine translator to show that TFMs are overparameterized. Across eight models from six families and 171 classification datasets, replacing half the transformer blocks with a single affine map costs at most 0.002 median AUROC, and isolating a single block remains competitive on most datasets. Per-block analysis shows why this is possible: in single-axis models no individual block is critical, and in dual-axis models the non-affine computation sits in a single block, while attention remains diffuse across depth. Unlike early-exit methods that discard the network's tail and require labeled data to train a new classifier, our translator succeeds by preserving the pre-trained head and accurately projecting intermediate states into its expected input space. Furthermore, at half-depth, a fitted map can be reused as-is across similar datasets (e.g., MIMIC-III and eICU), and for TabICL and TabPFN-3 it keeps most of its benefit even on unrelated datasets. These reductions remove up to 96% of the parameters with one block kept (1.5B for TabFM), speed up end-to-end GPU inference by up to $6\times$ at a median cost of 0.03 AUROC, and let a single-block TabPFN-3 serve cached queries $2.4\times$ faster with $6\times$ less memory at negligible cost.

Limitations and future work. Beyond demonstrating architectural redundancy, our analysis identifies specific conditions where extreme compression to a single block fails, such as maintaining continuous target rankings in regression, and for dual-axis models whose map cannot be absorbed (i.e., LimiX-16M), the per-feature map outweighs the removed blocks on very wide datasets. Additionally, our comparison between compression and shallow pre-training is currently limited to a 1.4M-parameter nanoTabPFN. Whether this performance parity holds at production scale, where pre-training shallow models from scratch is prohibitively expensive, remains an open question. These limitations motivate the design of adaptive architectures that allocate network depth only when strictly required. Given how replaceable most blocks are, a recurrent transformer that reuses one block's weights could supply that depth without the parameter count of current models. Finally, the window could be selected under a hardware budget instead of a fixed width, minimizing the calibration KL subject to a target memory or latency that also accounts for the cost of the map.

AI USE STATEMENT

In this work, we used generative AI tools for research execution: to implement well-specified components such as result loaders and table and figure generators, to write the driver scripts that run, monitor and relaunch the experiment sweeps, and to cross-check the numbers reported in the text against the saved result files. We did not use generative AI tools for research ideation, for the core ideas, or for formulating hypotheses. We also used AI-assisted search for retrieval and discovery, to find related and concurrent work, while we manually verified the discovered bibliography.

We have reviewed all AI-assisted work. For the code, every AI-written script was checked by the authors against values computed independently or against known results, and all numbers in the paper come from the saved result files rather than from AI output. For the text, AI was used for targeted polishing rather than wholesale generation, and every suggestion was evaluated and integrated by the authors so that the scientific content and the original intent were preserved. We take responsibility for the final content of this work, including any text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work introduces a method to compress existing pre-trained TFMs and collects no new data. Two of our evaluation datasets, MIMIC-III and eICU-CRD, contain de-identified electronic health records; we accessed them under the PhysioNet credentialed-access process and their data use agreements, and we do not redistribute them. All other datasets are public benchmarks. By enabling pre-trained models to run on modest hardware, such as a local clinical server or a CPU, the method lets sensitive tabular data stay on site rather than being sent to a remote service. Environmentally, our post-hoc compression avoids retraining: a compressed model is obtained from a closed-form fit on a few hundred unlabelled rows and then reduces inference compute and memory for every subsequent query. The primary ethical consideration of our specific method is its potential impact on the predictions for individual patients when a compressed model replaces the full one in a clinical setting. We examined this on the two ICU cohorts (Section B.10): at half depth the changes are small and uniform across record-completeness quartiles and outcomes, whereas with a single block kept the patients with the rarer outcome lose sensitivity. We therefore recommend the half-depth configuration for deployment, and any compressed clinical model should be re-validated on its target population before use.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Section 3 describes the translator, the ridge fit, the label-free window selection and the absorption of the map in full. Complete experimental details (datasets and splits, models and their ensembling settings, GBDT hyperparameter grids, and the exact hardware) are given in Section A, the nanoTabPFN training protocol in Section B.17, and the latency measurement protocol in Section 6 and Section B.16. The source code and the result files behind every table and figure will be released upon acceptance; the algorithmic description is complete enough for an independent implementation.

REFERENCES

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6679–6687, 2021.

Amir Rezaei Balef, Mykhailo Koshil, and Katharina Eggensperger. Is one layer enough? understanding inference dynamics in tabular foundation models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2605.06510.

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

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Mohamed Bouadi, Pratinav Seth, Aditya Tanna, and Vinay Kumar Sankarapu. Orion-MSP: Multi-scale sparse attention for tabular in-context learning. *arXiv preprint arXiv:2511.02818*, 2025.

Mohamed Bouadi, Pratinav Seth, Aditya Tanna, and Vinay Kumar Sankarapu. Orion-Bix: Bi-axial attention for tabular in-context learning. In *Proceedings of the ACM Web Conference 2026*, pp. 8673–8676, 2026.

D. Campos and J. Bernardes. Cardiotocography. UCI Machine Learning Repository, 2000. DOI: https://doi.org/10.24432/C51S4N.

Irene Cannistraci, Simone Antonelli, Emanuele Palumbo, Thomas M Sutter, Emanuele Rodolà, Bastian Rieck, and Julia E Vogt. Toast: Transformer optimization using adaptive and simple transformations. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=fSwMCsBtTG.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. Analyzing redundancy in pre-trained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4908–4926, 2020.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. arXiv:2506.16791.

Sebastian Felix Fischer, Liana Harutyunyan, Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – a curated tabular regression benchmarking suite. In *AutoML Conference 2023 (Workshop)*, 2023.

Google Research. TabFM (model release v1.0.0). https://github.com/google-research/tabfm, 2026. Checkpoint google/tabfm-1.0.0-pytorch.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35: 507–520, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. TabPFN-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations*, 2025.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Mozes Jacobs, Thomas Fel, Richard Hakim, Alessandra Brondetta, Demba E. Ba, and T. Anderson Keller. Block recurrent dynamics in vision transformers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=gH3HhnfWLC.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):160035, 2016.

Mykhailo Koshil, Matthias Feurer, and Katharina Eggensperger. TACTICL: Task-aware compression of tabular ICL models. In *International Conference on Automated Machine Learning (AutoML)*, 2026. arXiv:2608.10837.

Jaris Küken, Lennart Purucker, and Frank Hutter. Early stopping tabular in-context learning. In *ICML Workshop on Foundation Models for Structured Data*, 2025. arXiv:2506.21387.

Yanhong Li, Songlin Yang, Shawn Tan, Mayank Mishra, Rameswar Panda, Jiawei Zhou, and Yoon Kim. Distilling to hybrid attention models via KL-guided layer selection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RzbsHcFqIf.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=pIZxEOZCId.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36, pp. 76336–76369, 2023.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, 2025.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, 2019.

Donald Michie, David J. Spiegelhalter, and Charles C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994. Statlog (Heart) data: UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C57303.

Alexander Pfefferle, Johannes Hog, Lennart Purucker, and Frank Hutter. nanoTabPFN: A lightweight and educational reimplementation of TabPFN. *arXiv preprint arXiv:2511.03634*, 2025.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):180178, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847, 2025. URL https://openreview.net/forum?id=0VvD1PmNzM.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.11139.

Ross Quinlan. Thyroid Disease. UCI Machine Learning Repository, 1986. DOI: https://doi.org/10.24432/C5D010.

Dmitriy Shopkhoev, Ammar Ali, Magauiya Zhussip, Valentin Malykh, Stamatios Lefkimmiatis, Nikos Komodakis, and Sergey Zagoruyko. ReplaceMe: Network simplification via depth pruning and transformer block linearization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.02819.

Jack W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pp. 261–265, 1988.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

Aditya Tanna, Nassim Bouarour, Mohamed Bouadi, Vinay Kumar Sankarapu, and Pratinav Seth. Distilling tabular foundation models for structured health data. *arXiv preprint arXiv:2605.18702*, 2026.

The SUPPORT Principal Investigators. A controlled trial to improve care for seriously ill hospitalized patients: The study to understand prognoses and preferences for outcomes and risks of treatments (SUPPORT). *JAMA*, 274(20):1591–1598, 1995.

Shashank Venkataramanan, Amir Ghodrati, Yuki Asano, Fatih Porikli, and Amirhossein Habibian. Skip-attention: Improving vision transformers by paying less attention. In *International Conference on Learning Representations*, volume 2024, pp. 29654–29672, 2024.

William Wolberg. Breast Cancer Wisconsin (Original). UCI Machine Learning Repository, 1990. DOI: https://doi.org/10.24432/C5HP4Z.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Guri Zabërgja, Rafziq Kamel, Arlind Kadra, Christian M. M. Frey, and Josif Grabocka. End-to-end compression for tabular foundation models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, 2026. arXiv:2602.05649.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, Ningbo Dai, Renzhe Xu, Shuyang Li, Tianyang Zhang, Yue He, Yuanrui Wang, Yunjia Zhang, Zijing Xu, Dongzhe Li, Fang Gao, Hao Zou, Jiandong Liu, Jiashuo Liu, Jiawei Xu, Kaijie Cheng, Kehan Li, Linjun Zhou, Qing Li, Shaohua Fan, Xiaoyu Lin, Xinyan Han, Xuanyue Li, Yan Lu, Yuan Xue, Yuanyuan Jiang, Zimu Wang, Zhenlei Wang, and Peng Cui. LimiX: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025a.

Xiyuan Zhang, Danielle C. Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W. Mahoney, Cuixiong Hu, Huzefa Rangwala, George Karypis, and Bernie Wang. Mitra: Mixed synthetic priors for enhancing tabular foundation models. In *Advances in Neural Information Processing Systems*, 2025b. URL `https://openreview.net/forum?id=t8YRsWY6HM`.

A Implementation Details

A.1 Datasets

Table 2 lists the classification and regression datasets.

We use 68/12/20% train/calibration/test splits and three seeds (0, 1, 2) for every dataset, clinical and suite alike. Train, calibration and test splits are disjoint by index (test first, then calibration off the remainder), and patient-disjoint for the clinical cohorts (MIMIC-III, eICU-CRD) via `GroupShuffleSplit` on patient identifiers; encoders and scalers are fitted on the training rows only. A few standard benchmark datasets (e.g., Breast Cancer, Thyroid) contain a small number of intrinsically identical rows that can straddle splits by chance; these natural duplicates are exposed identically to the full and the compressed model, so they do not bias the relative compression cost. Numerical features are standardised on the training set; categoricals are integer-encoded; missing numericals are imputed with zero after scaling. The calibration split is also the set on which the translator is fitted. Not every model runs on every dataset. TabPFNv2, TabFM, both Orions and LimiX-16M support at most 10 classes and skip the six suite datasets with more classes, and Orion-MSP and LimiX-16M run out of GPU memory on a few large high-dimensional datasets (Orion-MSP and Orion-Bix also miss single seeds on a few of them for the same reason). These models

Table 2: Dataset summary. Suite rows aggregate benchmarks deduplicated by OpenML ID, each dataset counted once under one suite; clinical rows list each dataset individually.

Name	#Datasets	#Instances	#Features	#Classes
Benchmark suites (classification)				
OpenML-CC18 (Bischl et al., 2021)	56	500–96K	4–3,072	2–46
TabZilla (McElfresh et al., 2023)	44	148–1.0M	4–4,296	2–100
Grinsztajn (Grinsztajn et al., 2022)	25	3.4K–940K	7–419	2
TabArena (Erickson et al., 2025)	38	748–150K	4–1,776	2–8
Clinical (classification)				
Diabetes (Smith et al., 1988)	1	768	8	2
Heart (Michie et al., 1994)	1	270	13	2
Breast Cancer (Wolberg, 1990)	1	699	9	2
SUPPORT2 (The SUPPORT Principal Investigators, 1995)	1	9,105	46	2
MIMIC-III (Johnson et al., 2016)	1	26,538	74	2
eICU-CRD (Pollard et al., 2018)	1	30,000	74	2
Cardiotocography (Campos & Bernardes, 2000)	1	2,126	35	3
Thyroid (Quinlan, 1986)	1	3,772	21	3
Benchmark suites (regression)				
MIMIC-III length of stay (Johnson et al., 2016)	1	26,538	74	—
TabArena (Erickson et al., 2025)	13	907–54K	5–1,024	—
OpenML-CTR23 (Fischer et al., 2023)	27	517–72K	5–116	—
Total	212			

therefore run on 152–157 of the 163 suite datasets; all models run on the eight clinical datasets, and every comparison is between a model and its own full version on the same datasets.

A.2 Models

A.2.1 GBDTs Hyperparameters

XGBoost and LightGBM are tuned per dataset and seed by a 30-configuration random search on the calibration split over a discrete grid — `n_estimators` $\in \{300, 500, 800, 1200\}$, `max_depth` $\in \{4, 6, 8, 10\}$ (LightGBM: $\{-1, 6, 8, 12\}$, `num_leaves` $\in \{31, 63, 127, 255\}$, `min_child_samples` $\in \{10, 20, 40\}$), `learning_rate` $\in \{0.01, 0.03, 0.05, 0.1, 0.2\}$, `subsample` and `colsample_bytree` $\in \{0.6, 0.8, 1.0\}$ — with early stopping after 20 rounds, selected by calibration AUROC; they share test splits and seeds with the foundation-model rows. The default-hyperparameter variants are reported alongside in the suite rank tables.

A.2.2 Tabular Foundation Models

Table 3 summarizes the architecture of each model.

Ensembling. Prediction-time ensembling averages independent forward passes, which would mix per-block residuals across passes, so we reduce it to a single member where possible. TabPFNv2, TabPFN-3, Orion and TabFM run with one estimator (library defaults: four, automatic, 32 and 32). TabICL (v2 checkpoint) is the exception: its default 8-member ensemble runs in a single forward pass, so we keep it. TabDPT runs at its default configuration and LimiX at its default with a single inference-configuration member. Full and compressed models use the same setting, so this choice does not affect any Δ. To measure its cost, we refit each model with its library default on the eight clinical datasets, three seeds. Full AUROC rises by a mean of 0.002 (TabPFNv2, TabPFN-3), 0.001 (TabFM) and 0.006 (Orion-MSP, Orion-Bix); medians are at most 0.003. The largest gains are for the Orion models on the two ICU datasets ($+0.014$ to $+0.020$).

Effect of calibration rows on the full baseline. The compressed model uses the calibration split to fit the translator and to select the window, thus the full model never sees it. To analyze the

Table 3: Architecture of the eight evaluated models. The translator replaces blocks of the transformer stack only; encoders and head run unchanged. Params are the encoder and stack parameters (stack share of total), while context is the support-set cap in our experiments, not the model's limit.

		Encoders (never replaced)		Transformer stack (probed)			
Model	Tokenisation	Encoder	Params (M)	Blocks $\times$ d	Params (share)	Cached mode	Context
TabPFNv2	cell, dual-axis	linear (x and y)	0.01	12 $\times$ 192	7.1 M (98%)	yes	3,000
TabPFN-3	row	feature-distribution embedder + column aggregator	1.3	24 $\times$ 512	51.4 M (97%)	yes	3,000
TabICL	row	set transformer per column + 3-layer row transformer	1.3	12 $\times$ 512	25.7 M (93%)	yes	3,000
TabDPT	row	linear	0.08	16 $\times$ 768	75.6 M (98%)	no (retrieval per query)	1,500
TabFM	row	cell embedder + 2 column set transformers + 2 row interactors	19.5	24 $\times$ 2048	1,611 M (98%)	no	3,000
Orion-MSP	row	as TabICL + multi-scale row interactor + Perceiver memory	18.1	12 $\times$ 512	25.2 M (57%)	no	3,000
Orion-Bix	row	as TabICL	1.3	12 $\times$ 512	25.2 M (93%)	no	3,000
LimiX-16M	cell, dual-axis	mask-embedding linear	0.13	12 $\times$ 192	15.9 M (96%)	no	3,000

impact we refit every full model on train $\cup$ calibration (same support cap) and re-score the test split on the clinical datasets, three seeds. Only the five datasets below the cap can change (Diabetes, Heart, Breast Cancer, Cardiotocography, Thyroid): 117 runs (five datasets, eight models, three seeds; TabDPT on Thyroid already exceeds its 1,500-row cap). The median AUROC change is 0.000 for every model, per-model means range from -0.002 to $+0.003$, and the 16 runs that move by more than 0.01 split evenly (8 gains, 8 losses). Withholding the calibration rows does not impact the baseline.

A.3 Translator implementation and ablations

The following section provides additional details regarding Section 3.

Dual-axis models. For the dual-axis models (i.e., TabPFNv2, LimiX-16M), block outputs have shape $(n_{\text{items}}, n_{\text{features}}, d)$ and the map for feature index f is applied as $\mathbf{H}'[\cdot, f, \cdot] = \mathbf{H}[\cdot, f, \cdot]\, W_f + b_f$. With $n_{\text{features}} = 1$, the per-feature variant reduces exactly to the single-(W, b) case, so this is a generalisation of the shared linear translator rather than a different method. The number of feature tokens is set by each model's preprocessing (for LimiX, the frozen preprocessing below), so n_{features} is the token count, not the column count.

LimiX. LimiX refits its preprocessing on every `predict()` call on the concatenated rows, so the feature count and column identity drift between the calibration and evaluation passes, and a per-feature map would be applied to mismatched columns. We therefore fit the preprocessing once on the support rows and freeze it: categorical detection, the categorical encoder and the constant-column filter are fitted on the support set and only applied to the query rows, so the column layout is identical in every pass. With the frozen preprocessing the full model's predictions are identical on 7 of the 8 clinical datasets (-0.006 on Heart) and the per-feature map applies unchanged. On the 153 suite datasets both run, it loses 0.005 on average at Half (median 0.002, 83% of datasets within one point) and 0.052 at One-block (median 0.029), against 0.003 and 0.098 for TabPFNv2. A shared (W, b), which does not need the freezing, is near-lossless at width 3–4 but collapses with one block kept (-0.151 on the clinical datasets): the collapse reflects an incompatibility between the map and the preprocessing, not evidence that LimiX needs its depth.

Conditioning. Each ridge solve has $d+1$ unknowns per output column. Its rows are sequence positions from one forward pass over the support and calibration sets, capped at 500 (all 216 on Heart, the smallest dataset). The cap applies to positions only. TabPFNv2 and LimiX-16M fit one solve per feature token, each over the same $\leq$ 500 positions. TabICL fits one (W, b) across all 8 ensemble members (up to 4,000 correlated rows). The solve is therefore over-determined for TabPFNv2, LimiX-16M (d=192) and TabICL (d=512), slightly under-determined for TabPFN-3 and Orion (d=512), and under-determined for TabDPT (d=768) and TabFM (d=2,048). In the

under-determined cases the ridge penalty makes the fit well-posed, and the training residual ε becomes optimistic because the fit interpolates. All claims therefore rest on test AUROC, not on ε. The translator never sees test data or labels. Extraction and solve take 33–109 ms per window on one GPU.

Calibration budget. The translator is fitted on at most 500 positions of one forward pass. To test whether this is enough, we refit at the KL-selected windows on $N \in \{15, 30, 50, 100, 250, 500\}$ random subsets (three draws each, λ chosen label-free as always) on five clinical datasets and compare with the capped fit. For mild compression the budget is small: at width 3 every model is within 0.005 of the capped fit by N=100, TabFM (d=2,048) and Orion-Bix from N=15, TabICL, TabPFNv2, Orion-MSP and LimiX-16M from N=50, TabPFN-3 and TabDPT from N=100, because the KL-selected mild windows sit late in the stack, where representations change little. With one block kept the budget matters: TabDPT, TabFM and Orion-Bix need N=250 and TabPFNv2 the full 500, and at N=15 the one-block models fall 0.014–0.16 below the capped fit. Fitting on all positions of the pass still helps Orion-MSP ($+0.022$) and TabFM ($+0.010$), and at most 0.006 for the others. About 100 unlabelled positions therefore suffice for mild compression and 250–500 for extreme compression, far below the context the model already consumes (Figure 8).

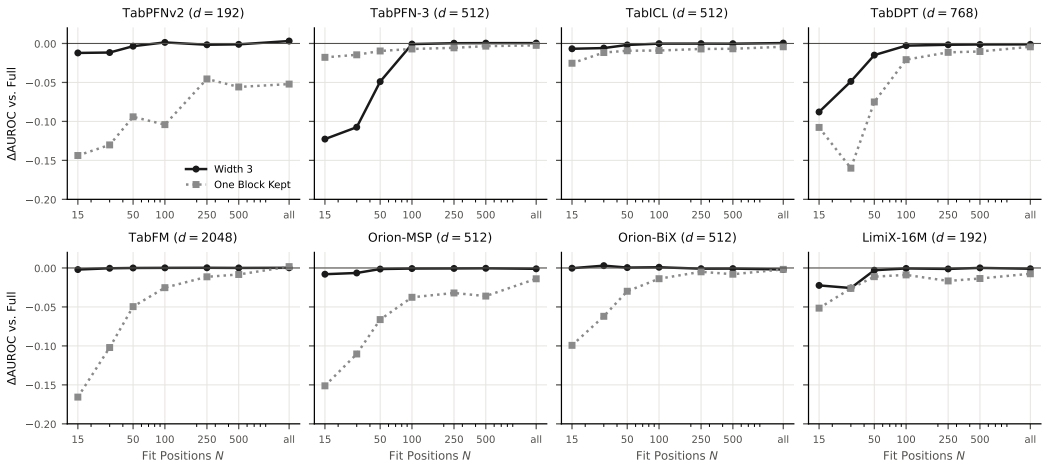

Figure 8: Calibration budget: ΔAUROC vs. number of fitting positions N (seed 0, five clinical datasets, three draws, KL-selected windows). Solid: width 3; dotted: one block kept. Mild compression needs at most 100 positions; with one block kept the knee moves to 250–500.

Per-feature vs. shared translator. The per-feature translator has one $d \times d$ map per feature token, $37{,}056$ parameters per token for TabPFNv2, against $589{,}824$ per TabPFNv2 block; the map therefore exceeds the blocks it replaces once the dataset has more than $16w$ feature tokens for a window of width w (64 tokens at width 4, 175 at one block kept). On MIMIC-III (39 tokens for 74 columns) the translator is already 61% of the four blocks it replaces at width 4, although the net parameter change at one block kept remains -71%; on the 14 suite datasets with more than 100 columns (61–302 tokens) it exceeds the four removed blocks on 12 of 14 and the eleven removed blocks on 6 of 14. A shared $d^2{+}d$ map would remove this dependence on the feature count, but it does not preserve accuracy (Table 4): at width 4 the per-feature map is within 0.02 of the full model on every dataset (median -0.002), whereas the shared map loses up to 0.15 (median -0.014; per-feature better on 10 of 14, worse on 1), and with one block kept the medians are -0.082 and -0.301 (per-feature better on 13 of 14). On wide datasets the dual-axis translator therefore measures redundancy but saves parameters only when the window starts at block 0, where the map folds into the encoder; the single-axis models, where the map is a single $d^2{+}d$ regardless of the dataset, have no such limit (262k parameters for TabICL against 23.6M removed at one block kept).

Gradient refinement of the translator. To test whether the closed-form translator can be improved with additional training, we refine it with 500 Adam steps of label-free final-representation matching through the frozen kept blocks. At width 4 this moves ΔAUROC by at most 0.003 and

Table 4: Per-feature vs. shared translator for TabPFNv2 on 14 suite datasets.

			translator	width 4		one block kept	
Dataset	columns	tokens	params (M)	per-feat.	shared	per-feat.	shared
cc18_1485	500	252	9.3	-0.018	-0.152	-0.321	-0.427
grin_45019	419	211	7.8	+0.003	-0.047	-0.080	-0.289
philippine	308	156	5.8	-0.009	-0.091	-0.154	-0.261
cc18_1501	256	184	6.8	-0.001	-0.021	-0.085	-0.416
cc18_40979	240	302	11.2	+0.000	-0.007	-0.056	-0.408
cc18_12	216	272	10.1	-0.000	-0.006	-0.090	-0.461
ta_46939	212	106	3.9	-0.013	+0.013	-0.047	-0.269
cc18_40670	180	92	3.4	-0.001	-0.003	-0.012	-0.472
volkert	180	185	6.9	-0.002	-0.068	-0.127	-0.322
ta_46908	170	86	3.2	-0.001	-0.001	-0.103	-0.012
jasmine	144	74	2.7	-0.016	-0.025	-0.024	-0.085
tz_40536	120	62	2.3	-0.002	-0.002	-0.013	-0.073
cc18_1486	118	61	2.3	-0.004	-0.004	-0.015	-0.121
ta_46980	111	101	3.7	-0.003	-0.056	-0.131	-0.313
median				-0.002	-0.014	-0.082	-0.301

calibration KL by at most 0.005 on any model (111 runs on the eight clinical datasets, three seeds); with one block kept the median change is 0.000 and the largest 0.036 (TabPFNv2 on Thyroid, where refinement moves further from Full). On regression at one block kept (MIMIC-III length of stay and the 13 TabArena regression sets, seed 0, R^2), refinement changes little for TabPFNv2, TabICL and TabFM (median gain ≤ 0.006) and matters only for TabDPT, recovering a median 0.07 of a median 0.45 loss. The residual gap is therefore not an optimization artifact of the closed-form estimator: gradient descent on the same objective leaves it essentially unchanged, which attributes it to approximation error, i.e., to the part of the removed blocks' computation that no affine map of the kept representation reproduces. Supervised fine-tuning on the task loss, as in TACTICL, optimizes a different objective with labeled data and falls outside the training-free setting studied here.

B Additional Results

B.1 Attention analysis

Measuring entropy. We measure attention diversity via per-head entropy: $\mathcal{H}(\mathbf{a}) = -\sum_j a_j \log a_j$, averaged across heads, queries, and samples. An entropy near $\log(S)$ (where S is the number of keys) signals uniform attention, while low entropy signals specialization. Because foundation models compute attention inside fused kernels that do not return weights, we recompute the softmax from the model's own projected queries and keys. We intercept the scaled-dot-product kernel during one forward pass and attribute each call to the block executing it. This interception includes the kernel's mask and scale, and applies TabFM's query-key normalization where relevant. We use the 500 unlabelled calibration rows as queries and all support rows as keys. We exclude the dual-axis models (TabPFNv2, LimiX) from this specific analysis because they interleave item and feature attention.

Top-k overlap between blocks. During the same forward pass, we record the five keys with the largest attention weight for every head across the 500 calibration queries. We then compare blocks using the Jaccard index of these key sets. We evaluate this on MIMIC-III and Higgs (seed 0), yielding 3,000 candidate rows (1,500 for TabDPT).

Matching heads by index across blocks yields a Jaccard index of just 0.02–0.04 for all evaluated models. Pooling the heads of a block (roughly 33 distinct keys per query) yields overall overlaps of 0.11–0.12 (TabICL), 0.09 (TabDPT), 0.11–0.12 (TabFM), and 0.06–0.08 (TabPFN-3). Consecutive blocks share slightly more (0.10–0.19), but this remains vastly different from the 0.005 expected from random sets. Notably, block 0 in TabPFN-3 shares almost no keys (0.01) with later blocks. Ultimately, the attended neighborhoods change at every block.

Comparison with a language model. For the NLP comparison in Figure 3, we evaluate six tabular TFMs on MIMIC-III and Higgs (using 3,000 support rows; 1,500 for TabDPT). We compare these against Qwen3-4B (Yang et al., 2025), a 36-layer language model with grouped-query attention (32 query / 8 KV heads), evaluated on WikiText-2 and IMDB. This 2-vs-2 dataset design ensures the contrast is not an artifact of any single dataset. Figure 9 shows all six models; the main text shows the three most representative.

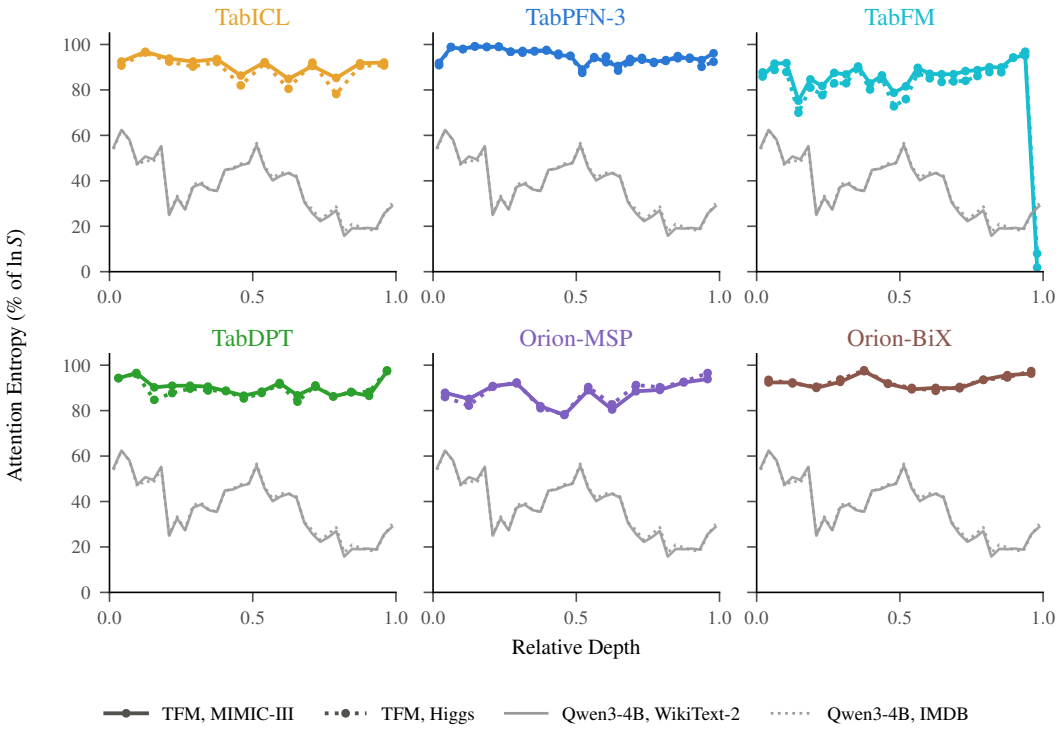

Figure 9: Per-block attention entropy as a percentage of the maximum possible entropy ($\ln S$) for all six tabular TFMs on MIMIC-III (solid) and Higgs (dotted) vs. Qwen3-4B (dashed). All models maintain near-uniform attention (70–99% outside TabFM's read-out block) compared to Qwen3-4B's sharply peaked attention (16–62%). TabFM's read-out block drops to 2–8%.

To compare models with different context lengths on one scale, we plot each block's entropy as a percentage of its maximum, $\ln S$ for S attended keys. The difference is not in the absolute entropy, which scales with sequence length, but in its profile across depth. Tabular models stay near-uniform at every block, whereas Qwen3-4B starts high, dips around layers 7–9, recovers in the middle, and falls again in the last quarter.

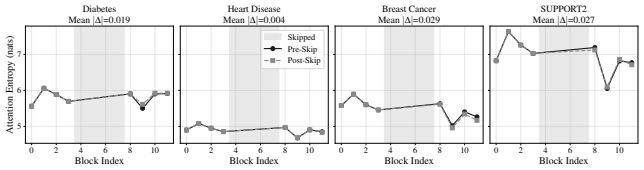

Figure 10: TabICL per-block attention entropy before vs. after replacing blocks 4–8 with the affine translator (shaded). Surviving blocks do not reorganise their attention to compensate.

Attention does not compensate for bypassed blocks. A direct test of the redundancy interpretation is whether surviving blocks reorganize their attention to compensate for the bypassed sub-stack. Figure 10 shows the per-block mean attention entropy before and after replacing a canonical window (blocks 4–8) in TabICL. A detailed per-dataset breakdown is provided in Table 5.

Table 5: TabICL attention stability: per-block entropy on surviving blocks before vs. after replacing a contiguous window. ε = translator residual; $|\Delta\mathcal{H}|$ = absolute entropy change (nats). Shifts are < 0.03 nats on average.

Dataset	Skip Window	ε	Mean $\lvert\Delta\mathcal{H}\rvert$	Max $\lvert\Delta\mathcal{H}\rvert$	Pre $\overline{\mathcal{H}}$	Post $\overline{\mathcal{H}}$
Diabetes	4–8	.197	.019	.114	5.80	5.82
Heart	4–8	.149	.004	.017	4.90	4.90
Br. Cancer	4–8	.122	.029	.097	5.48	5.45
SUPPORT2	4–8	.246	.027	.068	6.95	6.94
Diabetes (early)	2–6	.190	.028	.110	5.75	5.76
Heart (late)	6–10	.097	.001	.005	4.89	4.89

Across datasets, the per-block entropy changes by less than 0.03 nats on average, and at most 0.11 nats on any single surviving block (roughly 1–2% of the total entropy value). This indicates that the approximated sub-stack contributed negligibly to the downstream attention dynamics. Unsurprisingly, the shift is slightly larger for an early skip window (e.g., blocks 2–6 on Diabetes) because intervening closer to block 0 perturbs the input distribution to all subsequent blocks more heavily. Conversely, a late skip window (blocks 6–10 on Heart) causes almost zero shift (max $\Delta = 0.005$ nats). The surviving blocks therefore do not reorganize their attention, unlike the final layers of language transformers, which are thought to specialize to the training objective (Dalvi et al., 2020).

B.2 Per-block sensitivity, all models

Figure 11 extends Figure 4 to every reported model.

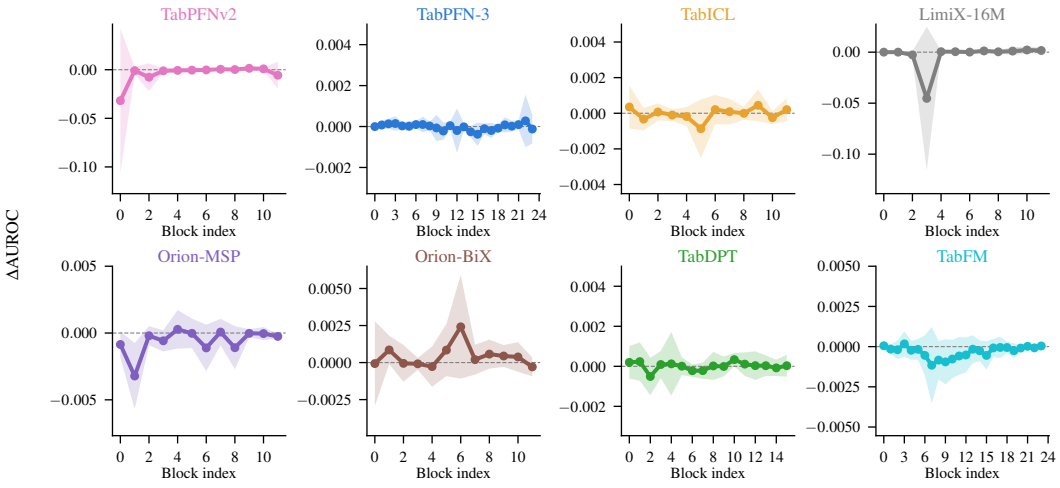

Figure 11: Per-block sensitivity of every reported model: ΔAUROC when a single block is replaced by the ridge translator (mean over the eight clinical datasets and three seeds).

B.3 Evaluation

B.4 Window selection

Window selection needs no labels. Every classification window is the one whose compressed model is closest in KL to the full model on the calibration split. No label is needed (the ridge λ is also chosen label-free), and because the head is kept, the KL costs nothing to compute. Table 6 compares this choice with the labelled one (best calibration AUROC), with the translator's residual ε, and with the test-set oracle. At Half, the KL matches or beats the labelled choice for every model except Orion-Bix (-0.001 vs. $+0.000$). ε never beats the KL: it is worse for six models and ties for the two Orions, because it ignores how the head amplifies errors and is not comparable across

depths. At One-block, where the only choice is keeping the first or the last block, the single-axis models stay within 0.002 of the labelled choice; for the two dual-axis models, the KL sometimes keeps the costlier block (Table 17).

Table 6: Window selection on the full sweep. Mean over datasets of the test ΔAUROC (averaged over seeds) (compressed $-$ full) of the window chosen by each criterion, per (model, dataset, seed): the smallest calibration KL $\mathrm{KL}(p_{\text{full}} \,\|\, p_{\text{c}})$ (ours, no labels; every result in the paper), the highest calibration AUROC (labelled), the smallest translator fit residual ε, and the highest test AUROC (oracle, an upper bound). All suite and clinical datasets the model runs on (n), three seeds, ridge translators.

	Half: mean ΔAUROC				One-block: mean ΔAUROC				
Model	KL (ours)	Labelled	ε	Oracle	KL (ours)	Labelled	ε	Oracle	n
TabPFN-3	$-.001$	$-.001$	$-.003$	$+.001$	$-.006$	$-.006$	$-.007$	$-.005$	171
TabICL	$-.003$	$-.003$	$-.004$	$-.001$	$-.015$	$-.015$	$-.016$	$-.014$	171
TabDPT	$-.003$	$-.004$	$-.005$	$-.000$	$-.024$	$-.025$	$-.025$	$-.023$	171
TabFM	$-.004$	$-.005$	$-.007$	$-.002$	$-.025$	$-.026$	$-.025$	$-.025$	165
Orion-MSP	$-.002$	$-.003$	$-.002$	$+.001$	$-.018$	$-.018$	$-.018$	$-.016$	160
Orion-Bix	$-.001$	$+.000$	$-.001$	$+.004$	$-.008$	$-.008$	$-.008$	$-.006$	165
TabPFNv2	$-.003$	$-.005$	$-.010$	$-.000$	$-.098$	$-.073$	$-.097$	$-.070$	165
LimiX-16M	$-.005$	$-.005$	$-.011$	$+.000$	$-.051$	$-.037$	$-.111$	$-.034$	161

Table 7: Half depth needs no window selection for the single-axis models. Mean over the suite datasets of the ΔAUROC (averaged over seeds) at Half (width $L/2$) with the window chosen by the calibration KL (no labels), with two fixed windows that need no selection at all (replace the last half of the stack; keep block 0 and replace the next $L/2$ blocks), and with the test-set oracle as the bound; in parentheses the share of datasets losing more than one point. Three seeds, ridge translators. TabPFNv2 (per-feature map) needs the selection.

Model	KL-selected	Fixed: last half	Fixed: keep block 0, replace next $L/2$	Test oracle	n
TabPFN-3	$-.001$ (4%)	$-.003$ (13%)	$-.007$ (26%)	$+.001$	163
TabICL	$-.003$ (10%)	$-.004$ (17%)	$-.008$ (27%)	$-.001$	163
TabDPT	$-.003$ (9%)	$-.006$ (23%)	$-.009$ (31%)	$-.000$	163
TabFM	$-.005$ (13%)	$-.006$ (13%)	$-.007$ (23%)	$-.002$	157
Orion-MSP	$-.002$ (7%)	$-.002$ (7%)	$-.012$ (33%)	$+.000$	152
Orion-Bix	$-.001$ (3%)	$-.001$ (4%)	$-.005$ (20%)	$+.004$	157
TabPFNv2	$-.003$ (9%)	$-.029$ (64%)	$-.124$ (79%)	$-.000$	157

KL-selected depth. The calibration KL can also choose how many blocks to replace. Table 8 applies the tolerance rule of Equation (3) ($\epsilon = 0.01$) on all 171 datasets, over the five swept widths, and on the eight clinical datasets, growing one window a block at a time. On all 171 datasets it removes 52–70% of the blocks of the single-axis models (42% for TabPFNv2, 22% for LimiX-16M) at a mean cost of 0.001–0.004 AUROC, with 85–97% of datasets within one point. Block by block on the clinical datasets it removes more for all models but TabPFNv2 (70–81% for the single-axis models, 50% for LimiX-16M, but 32% for TabPFNv2) at a similar cost, although greedy growth does not always find the lowest-KL window of each width.

B.5 Classification results

Figure 12 extends the left panel of Figure 1 to every model.

B.6 Per-dataset results and the identity control

Table 10 gives the per-dataset clinical results behind the clinical column of Table 9, with the **Id** rows: the One-block window with the translator replaced by the identity (naive layer dropping), which isolates the translator's contribution.

Table 8: Label-free depth from a drift tolerance. For each dataset and seed, the deepest compression whose calibration KL to the full model stays at most 0.01: on the 171 suite and clinical datasets among the swept widths $\{3, 4, L/2, L-2, L-1\}$ (KL-selected window per width), and on the eight clinical datasets along the greedy path that grows one window block by block. Removed = mean share of blocks replaced; ΔAUROC = mean test cost (compressed $-$ full); ≥ -0.01 = share of datasets within one point; greedy = min-KL = share of path steps where the greedy window is the lowest-KL window of that width. No label is used at any step. Three seeds.

	All datasets (5 widths)			Clinical (8 datasets, greedy)			
Model	removed	ΔAUROC	≥ -0.01	removed	ΔAUROC	≥ -0.01	greedy = min-KL
TabPFN-3	63%	-0.001	95%	70%	-0.001	100%	22%
TabICL	61%	-0.003	94%	81%	-0.002	100%	64%
TabDPT	52%	-0.003	91%	74%	-0.000	100%	46%
TabFM	53%	-0.002	95%	79%	-0.003	75%	81%
Orion-MSP	60%	-0.002	93%	73%	-0.001	100%	92%
Orion-Bix	70%	-0.004	85%	73%	+0.003	100%	74%
TabPFNv2	42%	-0.002	95%	32%	+0.001	100%	65%
LimiX-16M	22%	-0.001	97%	50%	-0.000	100%	56%

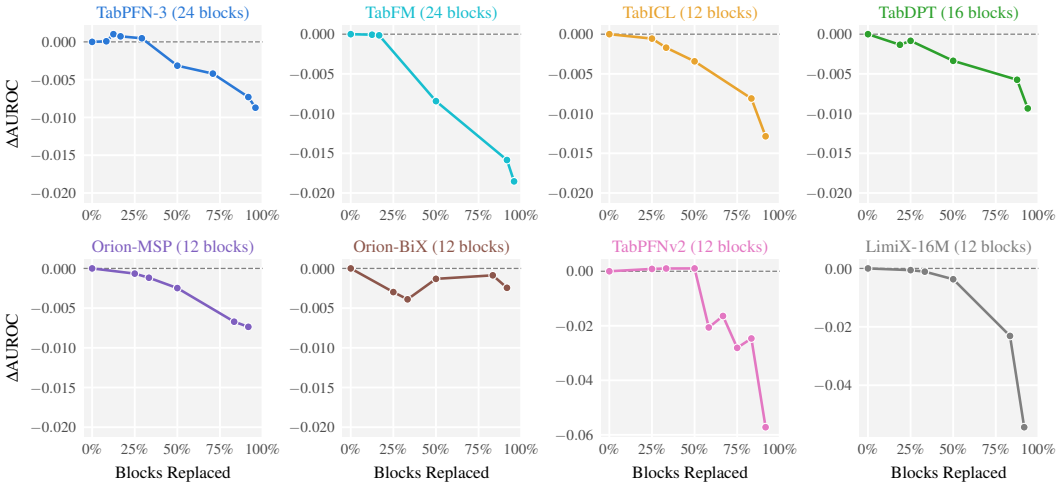

Figure 12: ΔAUROC vs. share of blocks replaced on MIMIC-III for every model, at every swept width (three seeds, KL-selected windows, ridge translator).

B.7 Suite-scale rank tables

Table 11 ranks every model over the suite datasets, with its full and Half versions ranked together (left) and the Half versions against tuned GBDTs (right). Replacing half the stack leaves the ordering of the models unchanged: the Half models rank in the same order as the full ones. The Half TabFM (average rank 3.53) is outranked only by the full TabFM (2.26) and beats every other full model. Against the trees, the Half versions of TabFM, TabICL and TabPFN-3 all rank above tuned LightGBM and XGBoost. Removing half the depth therefore changes neither which model is best nor how the models compare with trees.

Table 12 counts, per model, the datasets on which the compressed version wins, ties or loses against its own full version (a tie is $|\Delta\text{AUROC}| \leq 0.001$).

B.8 Check against TabArena

Since our protocol differs from the models' own evaluations (a single ensemble member where possible, a support set capped at 3,000 rows (1,500 for TabDPT), a 68/12/20 split per seed, and

Table 9: Classification results (AUROC). Clinical and benchmark suites, mean ± std over datasets and 3 seeds. Half: $L/2$ blocks replaced. One-block: $L-1$ blocks replaced. Ridge translators; windows selected by calibration KL (no labels). Id: the One-block window with the translator replaced by the identity (naive dropping). Per-dataset clinical results in Table 10.

Model		Clinical (8)	TabArena	TabZilla	CC-18	Grinsztajn
XGBoost		$.911_{\pm.084}$	$.860_{\pm.100}$	$.907_{\pm.086}$	$.911_{\pm.123}$	$.856_{\pm.096}$
LightGBM		$.909_{\pm.086}$	$.859_{\pm.100}$	$.907_{\pm.086}$	$.911_{\pm.124}$	$.857_{\pm.097}$
TabPFNv2	Full	$.904_{\pm.099}$	$.852_{\pm.112}$	$.901_{\pm.091}$	$.913_{\pm.122}$	$.834_{\pm.099}$
	Half (6/12)	$.904_{\pm.097}$	$.850_{\pm.113}$	$.896_{\pm.093}$	$.911_{\pm.122}$	$.829_{\pm.099}$
	One-block (11/12)	$.862_{\pm.103}$	$.792_{\pm.113}$	$.793_{\pm.137}$	$.777_{\pm.158}$	$.758_{\pm.091}$
	Id (11/12)	$.481_{\pm.083}$	$.503_{\pm.097}$	$.489_{\pm.080}$	$.521_{\pm.070}$	$.498_{\pm.065}$
TabPFN-3	Full	$.908_{\pm.093}$	$.861_{\pm.105}$	$.907_{\pm.089}$	$.913_{\pm.122}$	$.841_{\pm.098}$
	Half (12/24)	$.907_{\pm.093}$	$.860_{\pm.105}$	$.905_{\pm.089}$	$.913_{\pm.122}$	$.839_{\pm.098}$
	One-block (23/24)	$.906_{\pm.094}$	$.856_{\pm.105}$	$.898_{\pm.094}$	$.910_{\pm.123}$	$.834_{\pm.098}$
	Id (23/24)	$.880_{\pm.112}$	$.808_{\pm.105}$	$.823_{\pm.115}$	$.866_{\pm.130}$	$.775_{\pm.099}$
TabICL	Full	$.912_{\pm.089}$	$.863_{\pm.105}$	$.914_{\pm.087}$	$.919_{\pm.118}$	$.844_{\pm.096}$
	Half (6/12)	$.912_{\pm.090}$	$.860_{\pm.104}$	$.910_{\pm.087}$	$.918_{\pm.118}$	$.837_{\pm.098}$
	One-block (11/12)	$.909_{\pm.092}$	$.852_{\pm.102}$	$.891_{\pm.089}$	$.910_{\pm.120}$	$.822_{\pm.097}$
	Id (11/12)	$.839_{\pm.098}$	$.768_{\pm.109}$	$.763_{\pm.132}$	$.773_{\pm.130}$	$.745_{\pm.091}$
TabDPT	Full	$.902_{\pm.097}$	$.846_{\pm.113}$	$.887_{\pm.101}$	$.905_{\pm.125}$	$.823_{\pm.100}$
	Half (8/16)	$.903_{\pm.097}$	$.843_{\pm.113}$	$.881_{\pm.105}$	$.904_{\pm.126}$	$.819_{\pm.100}$
	One-block (15/16)	$.898_{\pm.098}$	$.821_{\pm.106}$	$.855_{\pm.101}$	$.887_{\pm.132}$	$.799_{\pm.101}$
	Id (15/16)	$.485_{\pm.071}$	$.496_{\pm.096}$	$.481_{\pm.115}$	$.497_{\pm.082}$	$.550_{\pm.083}$
TabFM	Full	$.914_{\pm.088}$	$.867_{\pm.102}$	$.926_{\pm.081}$	$.921_{\pm.116}$	$.875_{\pm.089}$
	Half (12/24)	$.912_{\pm.090}$	$.865_{\pm.103}$	$.919_{\pm.086}$	$.920_{\pm.118}$	$.865_{\pm.086}$
	One-block (23/24)	$.908_{\pm.092}$	$.852_{\pm.103}$	$.891_{\pm.089}$	$.910_{\pm.119}$	$.817_{\pm.109}$
	Id (23/24)	$.799_{\pm.097}$	$.688_{\pm.105}$	$.699_{\pm.122}$	$.710_{\pm.117}$	$.685_{\pm.098}$
Orion-MSP	Full	$.883_{\pm.100}$	$.837_{\pm.112}$	$.872_{\pm.108}$	$.899_{\pm.133}$	$.813_{\pm.106}$
	Half (6/12)	$.880_{\pm.099}$	$.834_{\pm.111}$	$.871_{\pm.106}$	$.897_{\pm.135}$	$.810_{\pm.106}$
	One-block (11/12)	$.870_{\pm.094}$	$.823_{\pm.108}$	$.848_{\pm.108}$	$.886_{\pm.134}$	$.795_{\pm.105}$
	Id (11/12)	$.755_{\pm.146}$	$.732_{\pm.133}$	$.722_{\pm.129}$	$.770_{\pm.117}$	$.741_{\pm.116}$
Orion-Bix	Full	$.886_{\pm.111}$	$.823_{\pm.129}$	$.858_{\pm.123}$	$.889_{\pm.143}$	$.796_{\pm.123}$
	Half (6/12)	$.888_{\pm.110}$	$.820_{\pm.134}$	$.859_{\pm.123}$	$.889_{\pm.142}$	$.795_{\pm.124}$
	One-block (11/12)	$.891_{\pm.106}$	$.810_{\pm.140}$	$.851_{\pm.119}$	$.883_{\pm.140}$	$.789_{\pm.123}$
	Id (11/12)	$.775_{\pm.151}$	$.704_{\pm.132}$	$.701_{\pm.146}$	$.737_{\pm.148}$	$.692_{\pm.139}$
LimiX-16M	Full	$.904_{\pm.095}$	$.855_{\pm.108}$	$.902_{\pm.099}$	$.914_{\pm.125}$	$.842_{\pm.095}$
	Half (6/12)	$.904_{\pm.095}$	$.852_{\pm.110}$	$.895_{\pm.102}$	$.912_{\pm.125}$	$.831_{\pm.100}$
	One-block (11/12)	$.888_{\pm.109}$	$.823_{\pm.105}$	$.826_{\pm.114}$	$.857_{\pm.132}$	$.806_{\pm.102}$
	Id (11/12)	$.565_{\pm.130}$	$.515_{\pm.097}$	$.498_{\pm.088}$	$.510_{\pm.092}$	$.499_{\pm.084}$

no bagging), in Table 13 we compare our Full AUROC (mean over three seeds) with the default-configuration entries of the public TabArena leaderboard (Erickson et al., 2025) on the 30 binary TabArena datasets, where the metric is the same and every dataset matches by name. TabArena evaluates each configuration on 3-fold outer cross-validation (repeated ten times below 2,500 rows, three times above) with the full training portion and cross-validation ensembling. Our numbers track the leaderboard per dataset (Pearson r 0.93–0.97) and sit 0.4–1.2 points below it on the datasets with at most 3,000 rows, where only the split protocol and ensembling differ, and 0.7–2.9 points below on the larger datasets, where our support cap also binds. Neither gap affects any claim in the paper, which compares each model with its own Full baseline under an identical protocol.

B.9 Decision metrics beyond AUROC

AUROC measures how well a model ranks examples, so we also check the decisions the compressed models make. Table 14 reports accuracy and balanced accuracy at both operating points. At Half, the compressed models make essentially the same decisions as the full ones: both metrics stay within about 0.01 of the full model on binary tasks and within 0.02 on multi-class tasks. With one block kept, an extreme setting, the decision metrics drop more than AUROC, mostly on multi-class tasks, where a small shift between the probabilities of similar classes can change the predicted class without changing the ranking. TabPFN-3 is the most robust (at most 0.03 balanced accuracy lost), and the largest drops come from the two dual-axis models, which are also the most affected in AUROC.

Table 10: Clinical datasets. AUROC for tuned GBDT baselines and eight tabular foundation models. **Full** = pre-trained model unchanged; Half = $L/2$ blocks replaced by the translator; One-block = $L-1$ of L blocks replaced; Id = the same maximal window with the translator replaced by the identity (naive layer dropping), isolating the translator's contribution. Translators: ridge with label-free λ selection; all skip windows selected label-free by the smallest calibration-split KL to the full model; subscripts $\pm$ std over seeds, where each seed is a different train/calibration/test split (patient-disjoint on MIMIC-III and eICU) and support subsample, so the Full rows vary with the seed as well. GBDTs are tuned (30-config random search on the calibration split) and share test splits and seeds with the foundation-model rows. All entries are raw AUROC.

Model		Diabetes	Heart	Br. Cancer	SUPPORT2	MIMIC	eICU	CTG†	Thyroid†
XGBoost		$.828_{\pm.022}$	$.885_{\pm.041}$	$.988_{\pm.008}$	$.981_{\pm.001}$	$.819_{\pm.005}$	$.792_{\pm.002}$	$.997_{\pm.001}$	$1.00_{\pm.000}$
LightGBM		$.819_{\pm.012}$	$.880_{\pm.036}$	$.990_{\pm.007}$	$.981_{\pm.001}$	$.817_{\pm.009}$	$.789_{\pm.004}$	$.997_{\pm.001}$	$1.00_{\pm.000}$
TabPFNv2	Full	$.844_{\pm.024}$	$.900_{\pm.030}$	$.995_{\pm.004}$	$.980_{\pm.000}$	$.773_{\pm.011}$	$.741_{\pm.004}$	$.998_{\pm.000}$	$1.00_{\pm.000}$
	Half (6/12)	$.841_{\pm.027}$	$.903_{\pm.036}$	$.994_{\pm.003}$	$.976_{\pm.001}$	$.774_{\pm.014}$	$.750_{\pm.003}$	$.996_{\pm.001}$	$.998_{\pm.001}$
	One-block (11/12)	$.826_{\pm.021}$	$.842_{\pm.047}$	$.995_{\pm.003}$	$.931_{\pm.005}$	$.716_{\pm.018}$	$.708_{\pm.007}$	$.979_{\pm.008}$	$.901_{\pm.014}$
	Id (11/12)	$.624_{\pm.047}$	$.314_{\pm.082}$	$.547_{\pm.026}$	$.503_{\pm.010}$	$.452_{\pm.007}$	$.455_{\pm.004}$	$.466_{\pm.023}$	$.488_{\pm.064}$
TabPFN-3	Full	$.842_{\pm.028}$	$.901_{\pm.029}$	$.995_{\pm.005}$	$.979_{\pm.001}$	$.792_{\pm.010}$	$.759_{\pm.005}$	$.998_{\pm.002}$	$.999_{\pm.000}$
	Half (12/24)	$.841_{\pm.028}$	$.899_{\pm.032}$	$.995_{\pm.005}$	$.977_{\pm.001}$	$.789_{\pm.009}$	$.760_{\pm.005}$	$.998_{\pm.001}$	$.999_{\pm.000}$
	One-block (23/24)	$.840_{\pm.023}$	$.905_{\pm.032}$	$.995_{\pm.005}$	$.975_{\pm.003}$	$.783_{\pm.009}$	$.758_{\pm.004}$	$.998_{\pm.002}$	$.999_{\pm.001}$
	Id (23/24)	$.797_{\pm.022}$	$.911_{\pm.037}$	$.991_{\pm.007}$	$.942_{\pm.005}$	$.715_{\pm.011}$	$.715_{\pm.006}$	$.974_{\pm.009}$	$.993_{\pm.002}$
TabICL	Full	$.845_{\pm.027}$	$.904_{\pm.035}$	$.994_{\pm.004}$	$.983_{\pm.001}$	$.800_{\pm.008}$	$.771_{\pm.007}$	$.999_{\pm.001}$	$1.00_{\pm.000}$
	Half (6/12)	$.846_{\pm.027}$	$.908_{\pm.037}$	$.994_{\pm.005}$	$.982_{\pm.001}$	$.797_{\pm.008}$	$.769_{\pm.007}$	$.999_{\pm.000}$	$1.00_{\pm.000}$
	One-block (11/12)	$.844_{\pm.023}$	$.911_{\pm.038}$	$.994_{\pm.004}$	$.975_{\pm.001}$	$.788_{\pm.010}$	$.764_{\pm.005}$	$.999_{\pm.000}$	$.998_{\pm.000}$
	Id (11/12)	$.799_{\pm.030}$	$.897_{\pm.042}$	$.988_{\pm.007}$	$.923_{\pm.002}$	$.708_{\pm.024}$	$.692_{\pm.033}$	$.893_{\pm.013}$	$.809_{\pm.102}$
TabDPT	Full	$.841_{\pm.027}$	$.898_{\pm.032}$	$.994_{\pm.005}$	$.969_{\pm.003}$	$.772_{\pm.003}$	$.750_{\pm.006}$	$.996_{\pm.005}$	$.997_{\pm.002}$
	Half (8/16)	$.844_{\pm.026}$	$.900_{\pm.035}$	$.995_{\pm.003}$	$.968_{\pm.003}$	$.769_{\pm.005}$	$.752_{\pm.006}$	$.998_{\pm.002}$	$.996_{\pm.002}$
	One-block (15/16)	$.842_{\pm.030}$	$.908_{\pm.036}$	$.995_{\pm.004}$	$.953_{\pm.005}$	$.763_{\pm.006}$	$.738_{\pm.012}$	$.995_{\pm.002}$	$.989_{\pm.002}$
	Id (15/16)	$.432_{\pm.031}$	$.611_{\pm.014}$	$.442_{\pm.027}$	$.456_{\pm.013}$	$.505_{\pm.012}$	$.471_{\pm.003}$	$.578_{\pm.011}$	$.386_{\pm.017}$
TabFM	Full	$.844_{\pm.020}$	$.909_{\pm.028}$	$.992_{\pm.006}$	$.984_{\pm.001}$	$.811_{\pm.009}$	$.771_{\pm.006}$	$.999_{\pm.001}$	$1.00_{\pm.000}$
	Half (12/24)	$.840_{\pm.024}$	$.912_{\pm.029}$	$.992_{\pm.004}$	$.983_{\pm.001}$	$.802_{\pm.008}$	$.765_{\pm.006}$	$.998_{\pm.003}$	$1.00_{\pm.000}$
	One-block (23/24)	$.840_{\pm.026}$	$.913_{\pm.025}$	$.992_{\pm.004}$	$.975_{\pm.001}$	$.792_{\pm.010}$	$.757_{\pm.008}$	$.998_{\pm.001}$	$.995_{\pm.001}$
	Id (23/24)	$.707_{\pm.033}$	$.864_{\pm.045}$	$.971_{\pm.026}$	$.885_{\pm.006}$	$.689_{\pm.010}$	$.690_{\pm.010}$	$.819_{\pm.038}$	$.766_{\pm.010}$
Orion-MSP	Full	$.831_{\pm.036}$	$.870_{\pm.053}$	$.995_{\pm.005}$	$.888_{\pm.007}$	$.756_{\pm.005}$	$.733_{\pm.005}$	$.999_{\pm.000}$	$.995_{\pm.002}$
	Half (6/12)	$.830_{\pm.035}$	$.871_{\pm.055}$	$.994_{\pm.006}$	$.875_{\pm.005}$	$.753_{\pm.003}$	$.732_{\pm.005}$	$.999_{\pm.001}$	$.987_{\pm.001}$
	One-block (11/12)	$.832_{\pm.033}$	$.875_{\pm.049}$	$.995_{\pm.004}$	$.859_{\pm.004}$	$.748_{\pm.004}$	$.730_{\pm.007}$	$.995_{\pm.000}$	$.928_{\pm.007}$
	Id (11/12)	$.809_{\pm.036}$	$.856_{\pm.038}$	$.987_{\pm.005}$	$.498_{\pm.029}$	$.673_{\pm.026}$	$.659_{\pm.012}$	$.881_{\pm.029}$	$.676_{\pm.019}$
Orion-Bix	Full	$.778_{\pm.044}$	$.870_{\pm.014}$	$.993_{\pm.006}$	$.973_{\pm.002}$	$.752_{\pm.007}$	$.727_{\pm.002}$	$.998_{\pm.000}$	$.997_{\pm.002}$
	Half (6/12)	$.790_{\pm.041}$	$.876_{\pm.013}$	$.994_{\pm.005}$	$.972_{\pm.002}$	$.751_{\pm.008}$	$.727_{\pm.004}$	$.998_{\pm.000}$	$.996_{\pm.002}$
	One-block (11/12)	$.816_{\pm.031}$	$.901_{\pm.032}$	$.994_{\pm.005}$	$.958_{\pm.004}$	$.750_{\pm.008}$	$.722_{\pm.006}$	$.998_{\pm.000}$	$.988_{\pm.001}$
	Id (11/12)	$.772_{\pm.023}$	$.873_{\pm.042}$	$.988_{\pm.006}$	$.618_{\pm.196}$	$.547_{\pm.058}$	$.619_{\pm.006}$	$.885_{\pm.010}$	$.896_{\pm.025}$
LimiX-16M	Full	$.837_{\pm.033}$	$.872_{\pm.017}$	$.995_{\pm.005}$	$.982_{\pm.001}$	$.789_{\pm.007}$	$.760_{\pm.006}$	$.999_{\pm.001}$	$.999_{\pm.000}$
	Half (6/12)	$.836_{\pm.021}$	$.881_{\pm.035}$	$.994_{\pm.005}$	$.979_{\pm.001}$	$.785_{\pm.006}$	$.760_{\pm.004}$	$.999_{\pm.000}$	$.999_{\pm.000}$
	One-block (11/12)	$.842_{\pm.020}$	$.885_{\pm.010}$	$.988_{\pm.004}$	$.977_{\pm.001}$	$.735_{\pm.055}$	$.707_{\pm.031}$	$.980_{\pm.005}$	$.988_{\pm.004}$
	Id (11/12)	$.541_{\pm.023}$	$.540_{\pm.061}$	$.875_{\pm.022}$	$.528_{\pm.005}$	$.588_{\pm.073}$	$.510_{\pm.115}$	$.550_{\pm.012}$	$.385_{\pm.021}$

† Macro-averaged OVR AUROC (CTG, Thyroid 3-class).

B.10 Subgroup analysis on the clinical cohorts

Does compression move the predictions of some patients more than others? We compare the per-row test predictions of the full and the compressed model on MIMIC-III and eICU (patient-disjoint splits, three seeds) at Half and One-block, at the KL-selected windows, along two axes a clinical deployment cares about: the completeness of a patient's record (the fraction of the 74 raw features observed before imputation; quartiles of the test rows, Q1 the least complete) and the outcome (Table 15). Decisions are compared at a common threshold, the full model's Youden-J threshold on the calibration split, so that a change in sensitivity or specificity reflects the compressed model's probabilities and not a re-tuned cut-off. At Half the change is small and uniform: for the six single-axis models and the two dual-axis models the ΔAUROC of every completeness quartile is within 0.007 of the overall value, sensitivity on the patients who died and specificity on those who survived move by at most 0.07 for the single-axis models (0.05 for TabPFNv2), and the mean absolute change of the predicted probability is 0.01–0.06 for the patients who died against 0.005–0.03 for those who survived. At One-block the cost lands where the tail of Section 5 predicts. Across completeness quartiles the single-axis models stay within 0.02 of their overall cost, while TabPFNv2 loses most

Table 11: Average rank over the shared suite datasets (lower is better, ± std; Friedman $p < 10^{-100}$). Left: Full and Half versions jointly ranked. Right: Half models vs. tuned GBDTs. Half-depth TabFM outranks every other full model.

Method	Avg. rank	±	Best	Mean AUROC
TabFM	2.26	2.48	101	0.9017
TabFM (Half)	3.53	2.54	6	0.8971
TabICL	3.88	2.35	10	0.8912
TabICL (Half)	5.29	2.26	4	0.8883
TabPFN-3	5.54	2.58	5	0.8868
TabPFN-3 (Half)	6.26	2.64	6	0.8858
LimiX-16M	7.74	3.38	1	0.8845
TabPFNv2	7.95	2.91	3	0.8814
LimiX-16M (Half)	9.67	2.95	2	0.8794
TabDPT	9.80	3.10	3	0.8720
TabPFNv2 (Half)	9.86	3.00	1	0.8781
TabDPT (Half)	11.50	2.95	0	0.8686
Orion-MSP	12.46	2.78	1	0.8622
Orion-Bix	13.23	2.89	1	0.8488
Orion-MSP (Half)	13.31	2.61	0	0.8603
Orion-Bix (Half)	13.72	2.86	0	0.8484

Method	Avg. rank	±	Best	Mean AUROC
TabFM (Half)	2.63	1.84	66	0.8971
TabICL (Half)	3.88	2.07	17	0.8883
TabPFN-3 (Half)	4.61	2.37	13	0.8858
LightGBM (tuned)	5.29	3.35	26	0.8882
XGBoost (tuned)	5.44	3.17	11	0.8882
LimiX-16M (Half)	6.91	2.44	2	0.8794
LightGBM	6.96	3.16	1	0.8820
TabPFNv2 (Half)	7.02	2.59	2	0.8781
XGBoost	7.73	3.17	2	0.8802
TabDPT (Half)	8.19	2.94	0	0.8686
Orion-MSP (Half)	9.59	2.49	1	0.8603
Orion-Bix (Half)	9.76	2.78	2	0.8484

Table 12: Per-model win/tie/loss of the compressed model (Half window, ridge) against its own full version over the suite datasets, with the median ΔAUROC.

Model	Win	Tie	Loss	Median ΔAUROC
TabPFNv2	14	48	95	-0.0017
TabPFN-3	19	77	67	-0.0004
TabICL	7	70	86	-0.0011
TabDPT	13	55	95	-0.0018
TabFM	10	47	100	-0.0021
Orion-MSP	22	44	86	-0.0014
Orion-Bix	27	72	58	-0.0004
LimiX-16M	23	33	97	-0.0019

on the *most* complete records (Q4 -0.048 against Q1 -0.015 on eICU, -0.057 against -0.035 on MIMIC-III). Across outcomes, the probabilities of the patients who died move about $2.5\times$ more than those of the survivors (0.03–0.10 against 0.02–0.04 for TabICL, TabFM, TabPFN-3 and TabDPT), and at the common threshold this is a loss of sensitivity at a gain of specificity: -0.17 and -0.12 (TabFM), -0.03 and -0.08 (TabICL), -0.02 and -0.05 (TabPFN-3) on eICU and MIMIC-III, against $+0.01$ to $+0.09$ specificity; the Orions and TabDPT move by at most 0.05. TabPFNv2's One-block predictions shift wholesale on MIMIC-III (sensitivity $+0.24$, specificity -0.64), a calibration change rather than a ranking one. LimiX-16M's One-block model loses 0.054 AUROC on both cohorts and its probabilities fall below the full model's threshold, costing 0.28–0.37 sensitivity for 0.11–0.19 specificity. The redundancy is therefore not paid for by a subgroup at Half, and at One-block the patients it costs are the ones with the rarer outcome, which is the clinical argument for Half as the deployable point. The outcome is the minority class (11–14% of rows).

B.11 Ablations: Early exit vs. translator

Since an affine map suffices to bridge the removed blocks, one could ask whether the class is already linearly decodable at the window start, so that the later blocks and the head add nothing a linear readout could not. We test this with an early-exit probe: the later blocks and the pre-trained head are discarded, and a logistic regression is fitted on the intermediate representations $\mathbf{H}_s$ with the support labels. If decodability at s were the whole story, this supervised probe should match the label-free translator. However, across 2,664 configurations over seven models (all but LimiX-16M) on the eight clinical datasets, the translator matches or beats the probe in 81% of cases. The probe's median cost stays near -0.02 at every depth (-0.023 early, -0.019 late), while the translator's shrinks from

Table 13: Our Full baselines vs. the TabArena leaderboard (default configurations, ROC AUC, 30 binary TabArena datasets): per-dataset Pearson correlation and the mean difference, split at 3,000 rows.

			$\leq$ 3,000 rows		$>$ 3,000 rows	
Model	datasets	Pearson r	n	ours − TabArena	n	ours − TabArena
TabPFN-3	30	0.951	9	-0.004	21	-0.022
TabICL	30	0.962	9	-0.005	21	-0.018
TabDPT	30	0.935	9	-0.004	21	-0.029
TabFM	30	0.968	9	-0.007	21	-0.018
LimiX-16M	30	0.933	9	-0.012	21	-0.007

Table 14: AUROC, accuracy and balanced accuracy at both operating points. Mean over datasets of Full / Half / One-block (KL-selected windows, ridge translator, three seeds; decisions by the most probable class, no recalibration), for the binary and the multi-class datasets (suite and clinical); n = datasets. At Half every metric stays close to Full; with one block kept the threshold-dependent metrics fall more than AUROC, most on multi-class tasks.

		AUROC			Accuracy			Balanced accuracy		
Model	n	Full	Half	One	Full	Half	One	Full	Half	One
Binary tasks										
TabPFNv2	114	.858	.856	.797	.840	.837	.771	.753	.749	.626
TabPFN-3	114	.865	.864	.860	.846	.844	.839	.761	.756	.746
TabICL	114	.868	.865	.853	.849	.846	.832	.765	.760	.733
TabDPT	114	.849	.847	.828	.833	.830	.813	.741	.734	.691
TabFM	114	.880	.876	.854	.859	.854	.829	.780	.767	.712
Orion-MSP	111	.838	.836	.822	.824	.822	.807	.730	.725	.682
Orion-Bix	114	.827	.826	.819	.818	.818	.811	.720	.717	.694
LimiX-16M	113	.864	.858	.831	.845	.839	.800	.762	.750	.662
Multi-class tasks										
TabPFNv2	51	.928	.925	.748	.826	.817	.479	.765	.747	.334
TabPFN-3	57	.940	.938	.932	.844	.840	.825	.789	.781	.759
TabICL	57	.947	.944	.932	.862	.855	.817	.811	.803	.747
TabDPT	57	.929	.924	.897	.823	.813	.725	.764	.750	.620
TabFM	51	.945	.940	.922	.867	.857	.802	.812	.794	.707
Orion-MSP	49	.925	.923	.904	.803	.795	.746	.743	.728	.642
Orion-Bix	51	.896	.895	.889	.780	.778	.753	.705	.700	.651
LimiX-16M	48	.929	.926	.837	.843	.835	.580	.779	.766	.449

-0.003 to $+0.000$. The two exceptions are consistent with Section 5: TabPFNv2's probe wins at early exits (-0.048 vs. -0.082), and Orion-Bix has the lowest win rate (63%) because its probe is already near-lossless (-0.010 to -0.005). The later blocks therefore do not refine an already-linear decision boundary; they move representations into the coordinates the pre-trained head reads, and the translator reproduces that move without labels.

B.12 One-block Ablations

Which block is kept. At One-block, the window keeps either the first or the last block, and the calibration KL chooses between them (Table 17). The single-axis models mostly keep the first block, which is also the cheaper option for them. TabPFN-3 costs the same either way, since its encoders do most of the work. The dual-axis models mostly keep the last block: keeping the first costs TabPFNv2 0.281 and LimiX-16M 0.230 AUROC, against 0.071 and 0.034 for the last. In the 8–25% of cases where the KL keeps their first block, it picks the costlier option.

Looping the kept block. To test whether the kept block contributes computation or only bridges to the head, we apply it $r \in \{1, 2, 3, 6, 12\}$ times, with the translator refitted on the looped output or removed (Table 18). With the translator, repetition brings no gain: the cost stays within 0.003

Table 15: Subgroup analysis on MIMIC-III and eICU (patient-disjoint, 3 seeds, windows selected by calibration KL). Per model at Half and One-block: overall ΔAUROC; ΔAUROC by feature-completeness quartile (Q1 = least complete); Δsensitivity/Δspecificity at the full model's Youden-J threshold; mean $|\Delta p|$ by outcome group.

		Half							One-block						
Model	Cohort	ΔAUROC	Q1	Q4	Δsens	Δspec	$\lvert\Delta p\rvert$ died	surv.	ΔAUROC	Q1	Q4	Δsens	Δspec	$\lvert\Delta p\rvert$ died	surv.
TabPFNv2	MIMIC-III	+0.001	+0.000	+0.004	+0.001	−0.011	0.053	0.029	−0.057	−0.035	−0.057	+0.243	−0.637	0.132	0.093
	eICU	+0.008	+0.013	+0.007	−0.048	+0.036	0.058	0.028	−0.033	−0.015	−0.048	−0.015	−0.166	0.107	0.073
TabPFN-3	MIMIC-III	−0.003	+0.002	−0.007	−0.022	+0.019	0.027	0.014	−0.009	−0.002	−0.014	−0.048	+0.042	0.043	0.023
	eICU	+0.002	−0.001	+0.003	−0.020	+0.018	0.022	0.010	−0.001	−0.006	−0.001	−0.024	+0.015	0.031	0.016
TabICL	MIMIC-III	−0.004	−0.003	−0.004	+0.006	−0.009	0.019	0.010	−0.012	−0.005	−0.012	−0.077	+0.038	0.063	0.025
	eICU	−0.003	−0.001	−0.008	+0.013	−0.014	0.016	0.008	−0.011	−0.007	−0.016	−0.031	+0.006	0.048	0.023
TabDPT	MIMIC-III	−0.003	−0.002	−0.001	−0.014	−0.002	0.025	0.013	−0.009	−0.008	−0.010	−0.031	+0.016	0.064	0.027
	eICU	+0.002	−0.005	+0.001	+0.007	−0.006	0.029	0.015	−0.013	−0.027	−0.016	−0.021	−0.012	0.075	0.030
TabFM	MIMIC-III	−0.008	−0.006	−0.010	−0.049	+0.023	0.045	0.018	−0.019	−0.012	−0.024	−0.122	+0.063	0.096	0.035
	eICU	−0.006	−0.009	−0.007	−0.071	+0.037	0.036	0.015	−0.014	−0.020	−0.017	−0.173	+0.087	0.075	0.027
Orion-MSP	MIMIC-III	−0.002	−0.000	−0.002	−0.015	+0.009	0.028	0.014	−0.007	−0.008	−0.004	−0.045	+0.027	0.058	0.027
	eICU	−0.001	−0.005	−0.002	+0.017	−0.011	0.025	0.013	−0.003	−0.021	−0.007	−0.025	+0.020	0.045	0.023
Orion-Bix	MIMIC-III	−0.001	+0.001	−0.002	+0.009	−0.008	0.015	0.008	−0.002	+0.008	+0.002	+0.010	−0.007	0.020	0.011
	eICU	−0.000	+0.003	−0.002	+0.005	−0.006	0.007	0.005	−0.005	−0.006	−0.004	+0.008	−0.020	0.009	0.007
LimiX-16M	MIMIC-III	−0.004	−0.004	−0.005	−0.031	+0.026	0.047	0.025	−0.054	−0.030	−0.052	−0.365	+0.192	0.175	0.069
	eICU	−0.001	−0.007	−0.000	−0.004	−0.005	0.035	0.018	−0.054	−0.108	−0.062	−0.282	+0.107	0.110	0.052

Table 16: Early-exit probe vs. affine translator: share of configurations where the translator matches or beats the probe, and median ΔAUROC vs. the full model across three depth bands (8 clinical datasets, 3 seeds, all widths). L = total blocks; ℓ = exit layer / translator start.

			early ($\ell \le L/3$)		mid		late ($\ell > 2L/3$)	
Model	runs	transl. ≥ probe	probe	transl.	probe	transl.	probe	transl.
TabPFNv2	288	67%	-0.048	-0.082	-0.035	-0.003	-0.016	+0.000
TabPFN-3	576	77%	-0.017	-0.002	-0.015	-0.002	-0.014	-0.000
TabICL	288	95%	-0.022	-0.001	-0.024	-0.000	-0.018	+0.000
TabDPT	384	86%	-0.032	-0.003	-0.032	-0.002	-0.030	-0.000
TabFM	552	91%	-0.026	-0.004	-0.018	-0.002	-0.031	-0.000
Orion-MSP	288	80%	-0.050	-0.006	-0.041	-0.002	-0.031	+0.000
Orion-Bix	288	63%	-0.010	-0.003	-0.005	-0.001	-0.006	+0.000
All	2664	81%	-0.023	-0.003	-0.022	-0.002	-0.019	+0.000

for TabICL and 0.006 for TabDPT, and grows from 0.006 to 0.016 for TabFM. Without it, no number of repetitions recovers the model (TabDPT stays near -0.4). The affine map, not additional computation from the kept block, is what makes One-block work.

Table 17: Which block is kept at One-block. Share of dataset–seed pairs whose KL-selected window keeps the first block rather than the last (suite / clinical datasets), and mean ΔAUROC when the first or the last block is kept (averaged over seeds; n: datasets with both windows).

Model	Keeps first block (suite / clinical)	Cost, first kept	Cost, last kept	n
TabPFNv2	14% / 8%	−.281	−.071	165
TabPFN-3	43% / 92%	−.007	−.006	171
TabICL	86% / 79%	−.016	−.019	171
TabDPT	97% / 100%	−.025	−.049	171
TabFM	93% / 96%	−.025	−.036	165
Orion-MSP	92% / 79%	−.017	−.023	160
Orion-Bix	83% / 83%	−.008	−.012	165
LimiX-16M	16% / 25%	−.230	−.034	161

B.13 CONTEXT SIZE

Figure 13 shows the context-size experiment behind Section 5. Across the suite, the same trend appears with dataset size: larger datasets lose more at One-block (Spearman $\rho = -0.22$ between the number of rows and the One-block cost, over the 163 suite datasets and eight models, $p < 10^{-14}$; -0.09 (LimiX-16M) to -0.34 per model).

Table 18: Looping the kept block r times, with the translator refitted on its output (+ map) or removed (no map). ΔAUROC vs. the full model, eight clinical datasets, three seeds.

Model	Variant	$r = 1$	$r = 2$	$r = 3$	$r = 6$	$r = 12$
TabICL	kept block $\times r$ + map	−.002	−.002	−.002	−.003	−.003
	kept block $\times r$, no map	−.067	−.034	−.034	−.044	−.071
TabDPT	kept block $\times r$ + map	−.005	−.005	−.004	−.005	−.006
	kept block $\times r$, no map	−.417	−.409	−.430	−.480	−.404
TabFM	kept block $\times r$ + map	−.006	−.014	−.015	−.016	−.016
	kept block $\times r$, no map	−.113	−.047	−.051	−.085	−.179

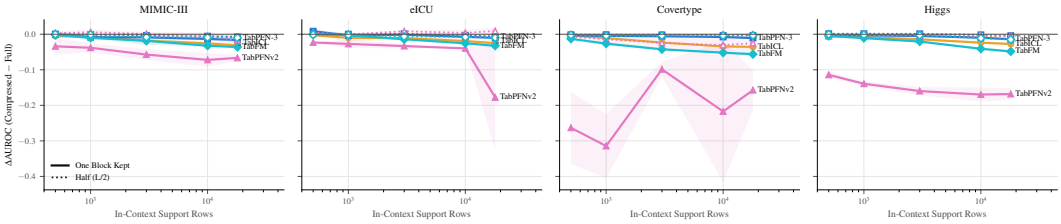

Figure 13: Context size vs. compression cost: ΔAUROC as the support set grows from 500 to 18,000 rows (3 seeds, shaded ± std; windows re-selected at every size). Solid: one block kept; dotted: Half ($L/2$). At Half the mean cost stays within 0.009 at every size (worst dataset: TabPFNv2 on Covertype, −0.030); with one block kept it grows with context (TabFM: −0.005 at 500 rows to −0.043 at 18,000).

B.14 Zero-shot map transfer across datasets

Because the shared affine map acts on the schema-free d-dimensional residual stream, a transformation fitted on one dataset can theoretically generalize to another. To evaluate this zero-shot transfer, we fit the map on a source dataset's unlabeled activations at the source's own KL-selected window and apply it unchanged to the target at that same window, so no target row is used to place or fit it. In Figure 14 we compare this transferred map against both a locally fitted map at the same window and a naive identity baseline that drops the blocks without a map.

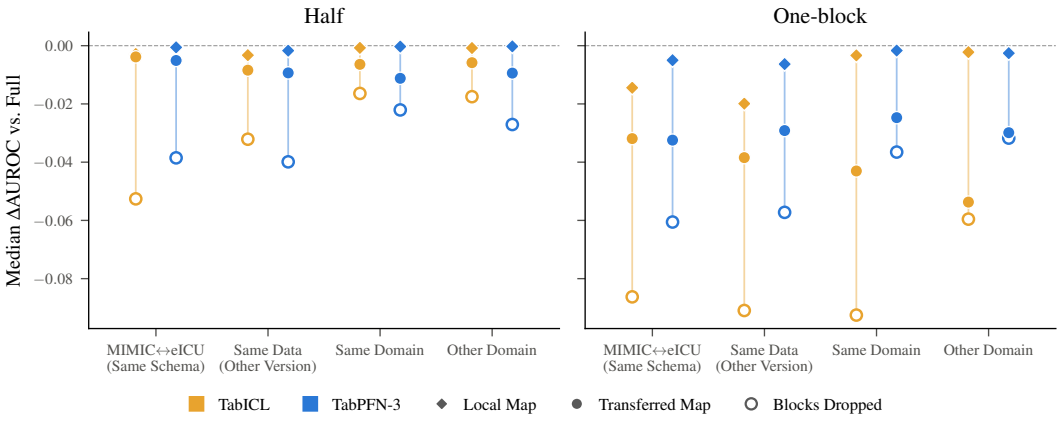

Figure 14: Reusing the map across sites and datasets. Median ΔAUROC vs. the full model on the target for the map fitted on the target itself (⧫), a map fitted on another dataset and applied unchanged (●, bar: interquartile range over pairs; for the two MIMIC-III↔eICU pairs, the range of the two), and dropping the blocks with no map (○). The closer ● lies to ⧫, the more of the local map survives transfer. Pairs are averaged over three seeds; all three maps sit at the source's KL-selected window. TabFM, TabDPT and a synthetic source are in Table 19.

For cohorts sharing an identical feature schema (e.g., the MIMIC-III and eICU intensive-care datasets), the affine map transfers well. At half-depth, a TabICL map fitted on one hospital cohort serves the other with an AUROC penalty of 0.004, against 0.003 for a local fit, so it keeps over 90% of the local map's benefit. However, across 446 disparate dataset pairs (out of 516 total source-target configurations), transfer weakens when the schemas differ, even within one domain (e.g., Heart and Diabetes, both medical but with different features and outcomes). On these pairs, a transferred half-depth map recovers only 60–65% of a local map's benefit for TabICL and TabPFN-3, about 30% for TabDPT, and none for TabFM (Table 19). Domain similarity does not help either: transferring maps from either medical or non-medical sources to MIMIC-III yields identical AUROC penalties for TabICL (-0.015 in both cases). With one block kept, the map transfers weakly, and better within a domain: for TabICL, a map borrowed from the same domain halves the loss of dropping the blocks (-0.043 vs. -0.093), while one from another domain barely improves on it (-0.054 vs. -0.060; Table 19). At Half, where domain gives no advantage, these results indicate that the skipped transformer blocks execute computations tied to specific features and outcomes, rather than general domain characteristics. A map can therefore be reused across sites or data versions that share a schema; a new schema needs its own local fit, which is fast and label-free. Table 19 reports the three quantities of Figure 14 for all four models in the transfer grid, with the synthetic source (a Gaussian input with a random two-layer teacher) as a fifth tier. Pairs: MIMIC-III↔eICU 2, same data 48, same domain 310 (254 for TabFM), other domain 136, synthetic 20; the other-domain pairs come from 9 sources (the clinical and credit datasets and two physics sets). For all four models a borrowed Half map transfers between versions of the same data and between MIMIC-III and eICU (TabDPT: -0.013 from eICU to MIMIC-III, against -0.002 local and -0.063 for dropping). Across different schemas TabICL and TabPFN-3 keep most of the local map's benefit, while a borrowed map recovers only part of the benefit for TabDPT (same domain: -0.040 vs. -0.050 for dropping the blocks) and none for TabFM (-0.012 vs. -0.011; its Half window costs only 0.01–0.02 even without a map). At One-block, a borrowed map falls far short of a local fit for every model, and the synthetic source breaks TabICL outright (-0.536, against -0.059 for dropping the blocks). The tiers do not share targets, so they differ in target difficulty as well as in similarity (note the different local and dropped columns). On the 16 targets that receive both same-domain and other-domain sources, the two tiers differ by a median of at most 0.002 at Half for TabICL and TabPFN-3.

B.15 Regression

Table 20 gives the per-width medians behind Figure 6, and Table 21 splits the One-block loss into a loss of scale and a loss of ordering.

For each regression model, dataset and width we re-ran the calibration-selected window of Table 20 at seed 0 and recorded, besides R^2, the Spearman rank correlation of the prediction with the target, the standard deviation of the compressed predictions relative to the full model's, and R^2 after an affine recalibration $y \approx a\,\hat{y} + b$ whose two coefficients are fitted on the compressed predictions of the calibration split (the regression analogue of the classifier's post-hoc scaling; test labels are never used). Table 21 gives the medians. At width 4 nothing moves. With one block kept the predictions are shrunk, the recalibration recovers a third of the loss where the outputs still carry the ordering (TabPFNv2, TabPFN-3, TabICL: on the datasets losing more than 0.05 R^2, the median fraction recovered is 0.33, 0.36 and 0.40) and little for TabDPT (0.05), whose ordering collapses, or TabFM (0.08), which loses little R^2 to begin with, and the rank correlation falls in the same order as the R^2 loss. LimiX-16M is the exception: its compressed predictions shrink to 0.22 of the full model's spread while the rank correlation falls only from 0.92 to 0.78, and the recalibration recovers three quarters of its loss (median ΔR^2 $-0.556 \rightarrow -0.139$). For the other models the loss is therefore mostly a loss of ordering, with the output scale a secondary and repairable part.

B.16 Extended hardware measurements

Absorbing the translator into the encoder or the head. Section 3 explains how the map can be absorbed by an adjacent linear layer. Encoder side (window starting at block 0): in TabPFNv2 each feature token of $\mathbf{H}_0$ is the output of a linear layer, so $\mathbf{H}_0\mathbf{W}_f + \mathbf{b}_f$ is again a linear function of that layer's input, and the per-feature map is absorbed into a per-feature copy of the encoder's last linear layer. Head side (window ending at block L): TabDPT's head reads $\mathbf{H}_L$ through a linear layer, which absorbs the map with its shape unchanged. Every other model has a normalization at the relevant

Table 19: Reusing the map across datasets, all models. Median ΔAUROC vs. the full model on the target, at the source's KL-selected window, for the map fitted on the target (local), a map fitted on another dataset (transferred) and no map (dropped), by source–target similarity. Pairs are averaged over seeds first: three seeds for TabICL and TabPFN-3, seed 0 for TabFM and TabDPT.

Model	Depth	Map	MIMIC↔eICU	Same data	Same domain	Other domain	Synthetic
TabICL	Half	local	−0.003	−0.003	−0.001	−0.001	−0.001
		transferred	−0.004	−0.008	−0.006	−0.006	−0.008
		dropped	−0.053	−0.032	−0.016	−0.018	−0.010
	One-block	local	−0.014	−0.020	−0.003	−0.002	−0.002
		transferred	−0.032	−0.038	−0.043	−0.054	−0.536
		dropped	−0.086	−0.091	−0.093	−0.060	−0.059
TabPFN-3	Half	local	−0.001	−0.002	0.000	0.000	0.000
		transferred	−0.005	−0.009	−0.011	−0.009	−0.007
		dropped	−0.039	−0.040	−0.022	−0.027	−0.029
	One-block	local	−0.005	−0.006	−0.002	−0.003	−0.002
		transferred	−0.032	−0.029	−0.025	−0.030	−0.034
		dropped	−0.061	−0.057	−0.037	−0.032	−0.032
TabFM	Half	local	−0.007	−0.006	−0.001	−0.002	−0.001
		transferred	−0.006	−0.008	−0.012	−0.010	−0.013
		dropped	−0.015	−0.021	−0.011	−0.004	−0.054
	One-block	local	−0.009	−0.025	−0.005	−0.006	−0.005
		transferred	−0.059	−0.081	−0.110	−0.089	−0.219
		dropped	−0.105	−0.171	−0.137	−0.126	−0.124
TabDPT	Half	local	0.000	−0.003	−0.001	−0.002	−0.002
		transferred	−0.025	−0.012	−0.040	−0.021	−0.036
		dropped	−0.122	−0.055	−0.050	−0.040	−0.024
	One-block	local	−0.013	−0.017	−0.012	−0.012	−0.012
		transferred	−0.118	−0.102	−0.260	−0.215	−0.202
		dropped	−0.264	−0.264	−0.433	−0.393	−0.398

Table 20: Regression over 41 datasets (MIMIC-III length of stay, 13 TabArena tasks, OpenML-CTR23): median full-model R^2 and median ΔR^2 (compressed − full) at the best of widths 3–4 (chosen by calibration R^2), at half depth, and with two and one blocks kept; medians over datasets of the mean over 3 seeds.

Model	datasets	Full R^2	best of 3–4	$L/2$	$L-2$	$L-1$
TabPFNv2 (12)	41	0.784	-0.001	-0.019	-0.176	-0.324
TabPFN-3 (24)	41	0.859	-0.002	-0.012	-0.111	-0.132
TabICL (12)	41	0.863	-0.003	-0.009	-0.065	-0.105
TabDPT (16)	41	0.769	-0.006	-0.026	-0.244	-0.401
TabFM (24)	41	0.873	-0.000	-0.001	-0.016	-0.022
LimiX-16M (12)	41	0.853	-0.002	-0.020	-0.293	-0.568

end, which prevents absorption: TabPFN-3, TabICL, TabFM and both Orions normalize the stack output before the head, and the encoders of TabPFN-3, LimiX and Orion-Bix end in a LayerNorm or RMSNorm. Across the full sweep, the selected One-block window touches the absorbing end in 86% of dataset–seed pairs for TabPFNv2 and 97% for TabDPT. Table 22 verifies that absorption is exact on the eight clinical datasets at three seeds (fp32 on the GPU): predictions with and without absorption differ by at most 4×10^{-6}, and every metric (AUROC, AUPRC, accuracy, balanced accuracy, F1, Brier, ECE) agrees to within 10^{-5}.

CPU deployment. Table 23 repeats Table 1 on a desktop CPU (Intel i9-14900K, fp32, batch of 512 queries, re-encode regime). There the transformer stack takes most of the forward pass for the lightweight-encoder models, so their end-to-end speed-up approaches the stack's (9.5–12.4× at One-block), while the heavy-encoder models stay at 1.2–1.8×.

Table 21: Scale vs. ordering decomposition (seed 0, medians over 41 datasets). Recal. = ΔR^2 after affine recalibration on the calibration split; ρ = Spearman rank correlation; scale = compressed/full std ratio. Most models lose ordering, not just scale.

Model	Blocks replaced	ΔR^2	recal.	ρ full $\rightarrow$ compr.	scale
TabPFNv2	4/12	-.002	-.000	.92 $\rightarrow$ .91	0.99
	11/12	-.333	-.164	.92 $\rightarrow$ .73	0.48
TabPFN-3	4/24	-.001	-.002	.93 $\rightarrow$ .93	0.98
	23/24	-.139	-.093	.93 $\rightarrow$ .86	0.72
TabICL	4/12	-.004	-.005	.93 $\rightarrow$ .92	0.98
	11/12	-.095	-.060	.93 $\rightarrow$ .86	0.71
TabDPT	4/16	-.007	-.005	.91 $\rightarrow$ .91	0.97
	15/16	-.445	-.337	.91 $\rightarrow$ .57	0.71
TabFM	4/24	-.000	-.000	.94 $\rightarrow$ .93	0.99
	23/24	-.023	-.023	.94 $\rightarrow$ .92	0.94
LimiX-16M	4/12	-.003	-.003	.92 $\rightarrow$ .92	0.98
	11/12	-.556	-.139	.92 $\rightarrow$ .78	0.22

Table 22: Absorbing the translator at One-block, eight clinical datasets $\times$ 3 seeds, fp32 on an RTX 5090. Pairs: dataset–seed pairs whose KL-selected window reaches the absorbing end (Heart keeps TabPFNv2's block 0 at two seeds). *Max* $|\Delta p|$: largest difference in predicted probability, with vs. without absorption. *Parameters*: translator size before absorption $\rightarrow$ extra parameters the absorbed model carries over the one-block model (TabPFNv2: per-feature copies of the encoder's last linear layer; range over datasets).

Model	Absorbed into	Pairs	Max $\lvert\Delta p\rvert$	Parameters
TabPFNv2	encoder	22/24	4×10^{-6}	0.22–1.45M $\rightarrow$ +5–43k
TabDPT	head	24/24	4×10^{-6}	0.59M $\rightarrow$ +0

B.17 NANOTABPFN TRAINING SETUP

Figure 15 shows the training dynamics across depths. Table 25 lists the values plotted in Figure 7.

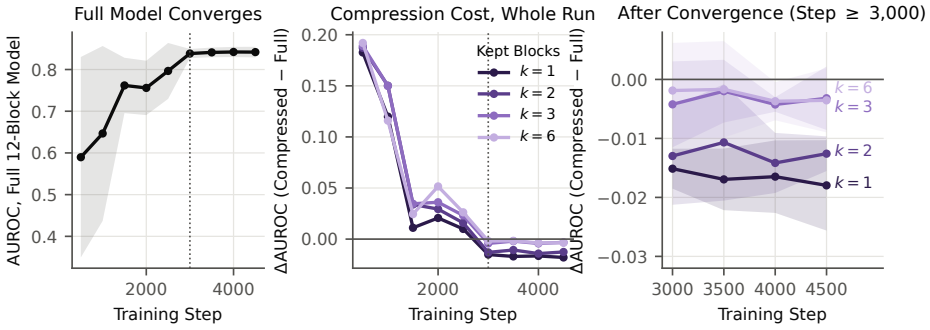

Figure 15: nanoTabPFN training dynamics: loss curves, evaluation AUROC and learning-rate restarts across depths $L \in \{1, 2, 3, 6, 12\}$ (three seeds each). Deeper models ($L \geq 6$) collapse to uniform predictions at the default learning rate and require an automatic restart at a quarter of the rate.

We use the nanoTabPFN reference code unchanged except for two additions. Every model has embedding size 96, 4 attention heads and an MLP hidden size of 192, from 132k parameters at L=1 to 1.37M at L=12. It is trained on a public set of pre-generated prior tasks (300k synthetic binary tasks, 150 rows $\times$ 5 features) for 5,000 steps at batch size 32 with schedule-free AdamW (no weight decay, gradient-norm clipping at 1), and the weights are taken at the optimizer's averaged point. The tasks are read in order, so every variant and seed sees the same sequence of prior batches and the seed changes only the initialisation. The first addition skips prior batches containing NaNs: the set

Table 23: **Hardware Impact on a CPU.** MIMIC-III on a desktop CPU (Intel i9-14900K, fp32). Blocks params: parameters of the transformer blocks kept, translator included. ΔFLOPs: forward-pass FLOPs removed. End-to-end: batch of 512 queries, context re-encoded on every call.

	Blocks params (M)			ΔFLOPs		End-to-end speed-up	
Model	Full	Half	One-block	Half	One-block	Half	One-block
TabPFNv2	7.1	5.0	2.0	−50%	−91%	2.2×	12.4×
TabPFN-3	51.4	25.9	2.4	−26%	−51%	1.2×	1.4×
TabICL	25.7	13.1	2.4	−23%	−42%	1.1×	1.3×
TabDPT	75.6	38.4	5.3	−49%	−92%	2.0×	11.0×
TabFM	1,611	810	71.3	−31%	−59%	1.3×	1.8×
Orion-MSP	25.2	12.9	2.4	−22%	−37%	1.1×	1.3×
Orion-Bix	25.2	12.9	2.4	−21%	−40%	1.1×	1.2×
LimiX-16M	15.9	11.4	4.8	−50%	−91%	2.0×	9.5×

Table 24: Stack-only vs. end-to-end speed-up on GPU (MIMIC-III, 3,000-row context (1,500 for TabDPT), one RTX 5090, batch of 512 queries, re-encode regime, median of 10 runs, CUDA synchronised at the stack boundaries). Stack: speed-up of the transformer stack alone; end-to-end: of the whole forward pass. The stack alone still gets close to $L/(\text{blocks kept})$ faster, but on a GPU it is a smaller share of the forward pass than on a CPU, so less of that gain reaches the end-to-end time.

		Width 4		Half		One-block	
Model	Stack share of time	stack	end-to-end	stack	end-to-end	stack	end-to-end
TabPFNv2	75%	1.5×	1.38×	2.0×	1.73×	11.2×	4.63×
LimiX-16M	91%	1.5×	1.43×	2.0×	1.83×	11.5×	6.10×
TabDPT	80%	1.2×	1.17×	1.7×	1.52×	6.9×	3.25×
TabFM	29%	1.2×	1.05×	2.0×	1.17×	23.1×	1.38×
TabPFN-3	19%	1.2×	1.02×	1.9×	1.09×	19.7×	1.23×
TabICL	10%	1.5×	1.02×	2.0×	1.04×	10.5×	1.09×
Orion-MSP	13%	1.5×	1.05×	1.9×	1.07×	9.6×	1.15×
Orion-Bix	26%	1.5×	1.09×	2.0×	1.15×	11.1×	1.31×

Table 25: **Pre-training shallow vs. compressing deep**, values of Figure 7. nanoTabPFN test AUROC, averaged over four clinical datasets; mean ± std over three seeds.

Kept blocks k	1	2	3	6	12
Trained at depth k	$.850_{\pm.009}$	$.902_{\pm.008}$	$.894_{\pm.005}$	$.902_{\pm.001}$	$.903_{\pm.003}$
Trained, compressed topology	$.852_{\pm.008}$	$.888_{\pm.008}$	$.900_{\pm.004}$	$.900_{\pm.002}$	—
Compressed $12 \to k$	$.854_{\pm.012}$	$.891_{\pm.001}$	$.903_{\pm.005}$	$.902_{\pm.007}$	—
Compressed $12 \to k$, prior-fitted map	$.428_{\pm.158}$	$.550_{\pm.116}$	$.832_{\pm.091}$	$.756_{\pm.202}$	—

contains five corrupt tasks, and the upstream NaN check is disabled, so a single such batch would ruin the weights. The second is a collapse check that uses the training loss alone, so no evaluation data is used during training. A run that falls into the uniform-prediction basin stays at the chance loss of the binary prior ($\ln 2 \approx 0.693$) for the whole run. If the mean training loss over the last 500 steps is not below $\ln 2 - 0.02$, the run is restarted from the same seed at a quarter of the learning rate. Starting from 4×10^{-3}, all depths $L \leq 3$ train at that rate at every seed, while $L \in \{6, 12\}$ collapse at every seed and train at 10^{-3}. No run needed a second restart. As a control, we also retrained the $L \in \{1, 2, 3\}$ models at 10^{-3} (the learning rate required for the 12-layer model). Their mean AUROC shifts by -0.002, -0.019 and $+0.007$, respectively; the $k = 2$ shift comes from one of the three seeds. The compressed models still match or slightly exceed these shallow models (0.854 vs. 0.849 at $k = 1$; 0.891 vs. 0.883 at $k = 2$; 0.903 vs. 0.901 at $k = 3$), so the learning-rate difference does not favor compression. The affine variant ("trained, compressed topology") inserts one 96×96 linear layer, initialised to the identity, after the first $\lceil L/2 \rceil$ blocks of an L-block model and trains it jointly. Its learning rates follow the same rule. For evaluation, each dataset is split per seed into a 100-row support, a calibration set of at most 150 rows and a test set of at most 1,000 rows. The five

features are the top-5 by F-score on the support. The 12-to-k compression uses the paper's ridge recipe, with the window selected on the calibration set.

B.18 CONTEXT EXTRAPOLATION IN NANOTABPFN

We repeat the comparison on the two large clinical cohorts, MIMIC-III and eICU. At a support of 100 rows the picture is unchanged. We then raise the support to 1,000 rows, about $7\times$ the 150-row tasks seen in pre-training, so the model must extrapolate in context length. Depth does not help here either: with $k \geq 2$ kept blocks the compressed models stay within 0.008 of the full 12-layer model on both cohorts (0.651–0.660 vs. 0.656 on eICU, 0.674–0.681 vs. 0.682 on MIMIC-III). Only the one-block model loses (0.660 on MIMIC-III, 0.630 on eICU). Shallow models trained from scratch do at least as well as the full model (0.660–0.667 on eICU and 0.694–0.701 on MIMIC-III for $k \geq 2$), and at $k = 2$ they lead the compressed models by 0.016 (eICU) and 0.027 (MIMIC-III).