Understanding the Surprising Generalization Properties of Tabular Foundation Models

Nour Shaheen[1,2,3,†], **Junwei Ma**[4,†], **Alex Labach**[5], **Frank Hutter**[6,7,8], **Valentin Thomas**[9], **Anthony L. Caterini**[5]

[1]Polytechnique Montréal, [2]Mila – Quebec AI Institute, [3]Chandar Research Lab, [4]University of Toronto, [5]Layer 6 AI, [6]Prior Labs, [7]ELLIS Institute Tübingen, [8]University of Freiburg, [9]Cohere

Abstract

Tabular Foundation Models (TFMs) increasingly rely on in-context learning, where a model receives labelled examples at inference time and predicts labels for new inputs without updating its weights. Existing TFMs are typically trained on either massive synthetic corpora or very large collections of real datasets. In contrast, we show that surprisingly strong transfer can emerge from self-supervised pre-training on just a *single* real table. In this setting, we also find that tables tend to be either broadly useful or broadly poor regardless of downstream prediction task, and that the strongest predictor of usefulness is the number of features rather than the number of instances. This leads to a task-centric interpretation of tabular pre-training: the number and the quality of tasks are essential for the pre-training of TFMs. We show that the same task-centric perspective can help corpus design at scale: fine-grained column-level pre-processing consistently improves downstream performance, while no improvements are observed when we filter or deduplicate at the dataset level. Finally, we offer a new perspective for how TFMs generalize: we believe that tabular in-context generalization is largely retrieval-based, and good models are those that learn to identify relevant examples in the provided context and aggregate them well. The mechanics of TFMs have been relatively understudied; our task-centric, retrieval-based perspective offers a new framework to guide future model and corpus design.

1 Introduction

Tabular data is commonly viewed as highly heterogeneous, leading to a prevailing belief that transfer learning across tabular domains is nearly impossible: e.g., *how could training on handwritten digits possibly transfer to estimating Californian housing prices?* However, recent tabular in-context learning (ICL) methods such as TabPFN [16, 17], TabDPT [22], and TabICL [27, 28]—termed tabular foundation models (TFMs)—have challenged this assumption. These methods train transformers using either massive amounts of synthetic data (millions of tables generated from structured priors [16, 17, 27, 28]) or by sampling from large collections of real tabular datasets [22, 12], randomizing context composition and prediction targets at each step to encourage in-context generalization. All modern tabular ICL models of which we are aware (e.g., [16, 17, 22, 27, 28, 34, 11, 31, 35]) share a similar task construction procedure during pre-training: a dataset is either selected or created, a column of this dataset is used as the target, and a subset of the remaining columns are used as features. Inference is then performed on new datasets while keeping the weights frozen: labelled examples from the evaluation dataset are given as context and the model predicts the values of unlabelled examples. The general belief is that, much like in large language models, only a very diverse pre-training set can cover the distribution of unseen tabular tasks and enable out-of-domain generalization [24, 16].

[†]Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

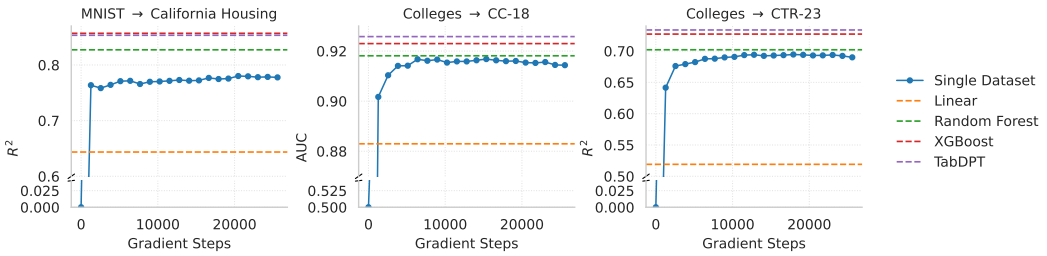


Figure 1: Transfer from a single pre-training dataset. **Left:** Training only on vectorized MNIST (treated as a table) and evaluating on CALIFORNIA HOUSING. **Middle and Right:** Training on the COLLEGES dataset and evaluating on the full CC-18 and CTR-23 evaluation suites, respectively.

Yet we find the opposite, and this surprising finding underpins our work: even with a *single* real-world table—such as vectorized MNIST [20]—a transformer trained with a naïve self-supervised learning (SSL) objective can still acquire generalization capabilities that transfer robustly *across domains*. Figure 1 illustrates this: a transformer trained from scratch only on the MNIST table yields strong in-context performance on structurally and semantically unrelated datasets like CALIFORNIA HOUSING [26]. Similarly, training on the COLLEGES dataset [14] and evaluating the resulting model on the OpenML CC-18 [3] and CTR-23 [9] benchmark suites (CC-BY license) ends up closely matching the performance of a random forest baseline.

In this paper, we first investigate which properties of single real-world datasets are most linked to generalization. We find: (1) Datasets that transfer well tend to do so universally across evaluation sets; (2) Feature count is a strong predictor of generalization, while instance count matters little; and (3) The number of *unique tasks* a model is pre-trained on strongly drives generalization ability.

We then apply this insight to pre-training on large data corpora. The task-centric view still applies with minor tweaks to control for data quality: more tasks generally help, as long as they have *sufficient quality*. In particular, removing duplicated, excessively correlated, or otherwise uninformative columns from each dataset further improves performance over baselines. Conversely, dataset-level filtering—even deduplication—is ineffective in our experiments. Although one might expect datasets resembling the "bad" single pre-training datasets to hurt performance, we find the opposite: adding them to "good" datasets still tends to improve generalization.

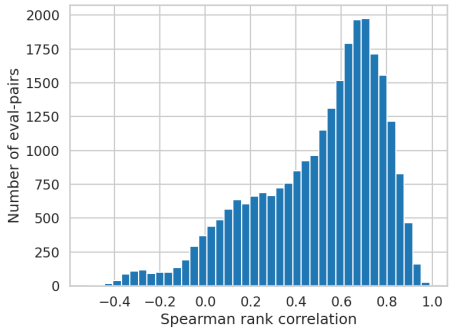
Finally, we provide an explanation for what makes generalization possible for TFMs. We believe the original Bayesian interpretation [24, 16] is unsatisfying since it requires downstream tasks to be covered by the distribution of the prior, which directly contradicts single-table generalization. We instead argue that TFMs act as learned retrieval-and-aggregation procedures. We provide experimental evidence for this by showing (1) a strong link between a model's performance and its ability to retrieve points within its own context, and (2) that architecture modifications enforcing nearest-neighbour-like behaviour have limited impacts on performance. We also show that the strongest single-dataset models display similar patterns in their attention maps, suggesting (although not proving) the existence of a universal retrieval space enabling generalization.

2 Generalization from single table training

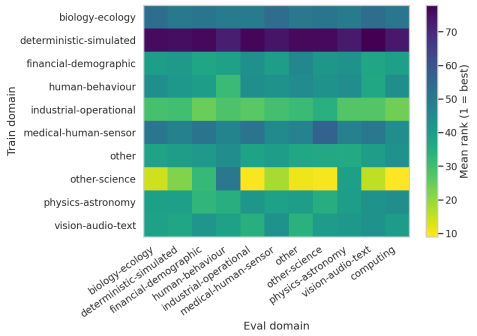
We design experiments to explore the central question of what constitutes a "good dataset" such that a tabular ICL model pre-trained on it can generalize. To isolate the effects of the training datasets themselves, we fix both the model architecture and the pre-training procedure throughout most experiments. Specifically, we use the shared backbone architecture from TabPFNv1 [16] and TabDPT [22], a widely adopted design for tabular ICL. We use the pre-training procedure from TabDPT to train tabular ICL models from scratch using a single real dataset only; as explained above, this procedure is fundamentally similar to all standard TFM training protocols. To isolate the effect of data, we choose *not* to use retrieval mechanisms (as used in TabDPT) during pre-training in this section, although it is enabled during inference. We evaluate the generalization performance on established benchmarks in order to easily compare with reported baseline performances directly.

2.1 Are good pre-training datasets universally good?

In Figure 1, we showed the surprising finding that pre-training an ICL tabular model on MNIST can yield strong performance on unrelated datasets. We ask: *"Is each pre-training dataset only good for some downstream task or does it generalize universally?"* We pre-train models separately on each of $N_{\text{train}} = 88$ training datasets from the TabDPT pre-training data, spanning different modalities and sizes, and evaluate the resulting 88 models on $N_{\text{eval}} = 107$ diverse datasets: the 72 CC-18 [3] classification datasets and 35 CTR-23 [9] regression datasets. No dataset appears in both pre-training and evaluation. Linear and RandomForest baselines were computed using `scikit-learn` defaults, except with logistic regression maximum iterations increased to 1000. XGBoost is the tuned baseline from McElfresh et al. [23], and TabDPT is computed from the public repository (Version 1.1, Apache-2.0 license).



(a) Ranking Correlation: The high values indicate that pre-training datasets which are good on one evaluation dataset tend to be good on others too.



(b) Domain to Domain Transfer: We do not observe stronger transfer when the pre-training and evaluation domain are shared.

Figure 2: Universality of dataset quality. **(a)** We compute the rank of training sets for each evaluation dataset and then plot a histogram of the Spearman correlation between the ranks on all pairs of evaluation datasets. Most evaluation datasets have very correlated ranks. **(b)** We group the training and evaluation datasets into distinct domains and plot average ranks (lower is better). Training and evaluation pairs from the same domain do not appear to transfer better than ones from different domains.

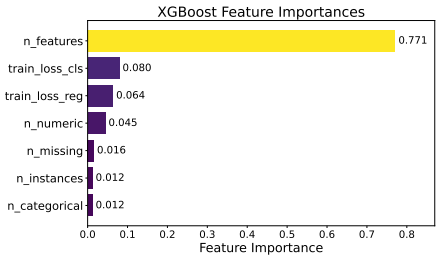
In Figure 2a, for each evaluation dataset, we rank the pre-training datasets based on their performance on that task, leading to N_{eval} rankings. We compute Spearman correlations over all ranking pairs, with higher values indicating stable relative rankings. Since the histogram is heavily left-skewed, it indicates that if a dataset is good for one evaluation dataset it tends to be good for others too, *even across vastly different evaluation tasks*. Next, we investigate whether a training dataset tends to generalize better to evaluation datasets within the same *domain*. We manually categorize both pre-training and evaluation datasets into domains, record the final performance (either accuracy for classification or correlation for regression) for each single training dataset, and then rank the training datasets from best to worst for each domain. Finally, we aggregate over the domains of the training datasets. If datasets from, e.g., Domain A are expected to transfer, on average, better to Domain A, we would expect a strong diagonal dominance in this matrix. However, in Figure 2b, we can see this is not the case: within-domain generalization is not higher. Rather, across the board, some domains are just broadly better for pre-training (e.g., `other-science` is great, `deterministic-simulated` is not).

2.2 What constitutes a good dataset?

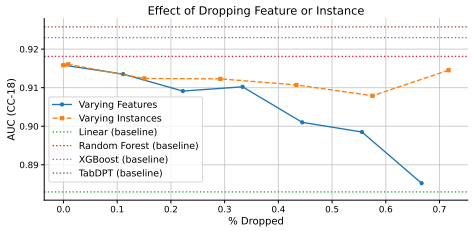
To understand which properties of a training dataset yield successful generalization, we run a meta-analysis on the N_{train} pre-training datasets. For each dataset, we collect descriptive meta-features, including: the number of features, number of instances, number of categorical features, number of numeric features, amount of missing values, and final pre-training losses for both classification and regression heads. These meta-features are split into meta-train (80%) and meta-test (20%) sets.

Using these meta-features, we train an XGBoost [6] regressor to predict a dataset's average downstream generalization score, defined as the average of correlation for regression tasks and accuracy for classification tasks. The resulting model achieves an R^2 of 0.67, indicating that roughly

two-thirds of the variance in generalization can be reproduced from straightforward dataset properties alone. Notably, when evaluating feature importance (see Figure 3a), the number of features emerges as by far the strongest predictor of downstream transfer performance. In contrast, the number of instances—often presumed critical in classical settings [33]—shows negligible predictive power. This suggests that the richness of the feature space is far more important than purely the dataset size for tabular ICL-based generalization. We validate this in Figure 3b by removing a fraction of either rows or columns on the COLLEGES dataset, showing that features tend to matter much more than instances. After dropping about 70% of features, the model performance drops to a similar level as a linear model, while dropping more than 70% of instances still allows the model to maintain performance close to a random forest. We see a similar phenomenon across many other datasets.



(a) Feature importance from an XGBoost model trained to predict average performance across 107 downstream tasks from dataset metadata.



(b) Column vs. Row Importance: We analyze the downstream performance when removing either features or instances from the COLLEGES dataset.

Figure 3: Not all cells in a table are made equal: the number of features matters much more than the number of instances as pre-training dataset for tabular ICL models.

2.3 Training on many good tasks unlocks generalization

To understand why dropping features caused a strong performance drop, we revisit the randomized procedure used for pre-training: a column is selected as target while a subset of other columns are used as features; we call this a *task*. Using a fixed supervised target provides just one task, while using a randomized target and features can have $\mathcal{O}(k\ 2^k)$ tasks for k columns. Thus, the role of columns may be more important than the role of rows as it is directly tied to the number of tasks or feature relationships the model is exposed to during training.

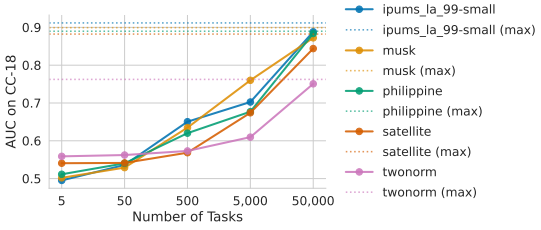


Figure 4: Downstream AUC vs. the number of unique tasks used during training. (max) denotes the unrestricted setting where the sampler draws from the full set of eligible task configurations with no upper count limit. More tasks during pre-training consistently leads to better transfer.

We disentangle the impact of the number of tasks from the number of features in Figure 4. We do not change the amount of instances or features in a given dataset but instead we vary the number of tasks the model is allowed to pre-train on. To be precise, we select N tasks using allowed permutations (one target, the remaining columns being either masked or used as features) and vary N for several datasets. A striking pattern emerges: across all datasets, increasing the number of tasks from 5 to tens of thousands drives the performance from barely above random (AUC $= 0.5$) to good. Thus, although inherent dataset differences matter and some datasets are significantly better for pre-training, the main factor explaining cross-domain generalization is the number of tasks seen during pre-training. In other words, the key is not the amount of pre-training data in tokens or bytes of memory, but the number of tasks/feature-target relationships that enable a tabular ICL model to generalize.

We further support the task-diversity view by training models on pairs with one dataset from the top quartile and one from the bottom quartile of Table 3. Pairing a strong dataset with a weak one does not hurt performance and can even slightly improve it: COLLEGES (AUC $= 0.918$ alone) with ANALCATDATA_SUPREME (0.814 alone) reaches 0.919, and APSFAILURE (0.918 alone) with POKER-HAND (0.777 alone) also reaches 0.919. These results suggest that datasets performing poorly during single-table pre-training are not intrinsically uninformative: they may be task-poor, but their tasks can complement those of a stronger dataset. This pattern also reappears in the next section.

3 From one to many: pre-training insights for large corpora

The preceding section showed how pre-training data properties such as feature count, sample size, and task diversity affect generalization in the single-dataset regime. We extend these observations to the practical setting where a TFM is pre-trained on a large, heterogeneous corpus of real-world tabular datasets. We ask the question of how to curate and pre-process such a corpus. We ablate along two axes: (i) **dataset-level selection:** which datasets to include in the corpus, and (ii) **feature-level pre-processing:** how to filter the features of each dataset prior to training. Our key findings are that exact deduplication is unnecessary, aggressive feature-based dataset-level filtering hurts performance, and lightweight column-level pre-processing is consistently beneficial.

We use part of the datasets collected by Hosseinzadeh et al. [18] as our pre-training corpus: $1{,}732$ datasets collected from OpenML [4], excluding datasets from the CC-18, CTR-23, and TabArena benchmarks [8], which we use for evaluation. We use a training context length of $1{,}024$ (512 queries) and an inference context length of $2{,}048$, and do not pre-process evaluation datasets. Our main evaluation metric is Interquartile Mean (IQM), which is standard and robust to outliers [1, 22]. We defer the full experimental setup, including model and training details, to Appendix B.1.

3.1 Does dataset-level selection help?

Two natural corpus-curating strategies would be deduplication and filtering out "universally bad" datasets. We find that neither is beneficial. **Deduplication:** Although deduplication benefits LLM training [21], exact deduplication is unnecessary here: models trained on the full corpus ($1{,}732$ datasets) and a deduplicated corpus ($1{,}535$ datasets, identified by the SHA-256 hash of X and y) are statistically indistinguishable on CC-18 and CTR-23 (Table 1). We attribute this to the SSL objective, which already reduces the impact of duplicates through random column-target sampling and independent context windows at each step. **Dataset-level filtering:** A reductive direction here would be to restrict the pre-training corpus to feature-rich datasets, since we established that those generalize better than low-feature ones. However, we previously showed that even a table with very few features can induce non-trivial out-of-domain generalization. Low-feature datasets are qualitatively different from high-feature ones, covering simpler dependency structures, smaller column-count regimes, and distinct domains. We test this by restricting both the full and deduplicated corpus to datasets with at least 30 features (607 datasets, and 557 after deduplication). Table 1 shows that this degrades performance across the board, with similar performance regardless of deduplication: excluding low-feature datasets reduces the number of diverse tasks in a way not offset by higher average per-dataset quality.

3.2 Column-level pre-processing as a better alternative

Since coarse dataset filtering is not fruitful, we instead perform fine-grained feature cleaning to increase task quality and diversity. For each dataset, we apply a subset of three operations, each ablated individually by training a separate model for every combination. **NaN-column drop** removes columns where more than 50% of values are missing; after TabDPT's mean imputation, such columns become nearly constant, so dropping them improves task quality. **Correlation deduplication** greedily removes columns until no pair has absolute correlation above 0.90, improving task diversity. The target y is included in the joint correlation matrix so features near-collinear with it are also removed, while y itself is protected from removal. We test Pearson and Spearman variants. At each greedy step, the column with the greatest number of above-threshold correlation partners is removed, with ties broken by lower variance. All columns are treated as numeric: categorical features are pre-encoded as integers and used as-is. **Minimum-feature enforcement** (mf-k) skips any dataset for which fewer than k feature columns remain after the above steps; we set $k = 5$ by default. Unlike the two column operations above, mf-k is a dataset-level exclusion rule rather than a column transformation: it is applied *after* column cleaning to drop tables left with too few usable columns. It's a light cleaning step that preserves most of the corpus rather than an aggressive dataset selection rule that discards whole task pools.

Table 1 reports the full ablation. Every pre-processing variant outperforms the baseline on regression, with IQM R^2 gains from $+0.002$ to $+0.014$. Classification gains are smaller but broadly consistent: all variants improve accuracy, and most improve AUC. This is consistent with the single-dataset findings: generalization is governed by the number of unique, good tasks the model is pre-trained on. Dataset filtering discards entire task pools, whereas column pre-processing removes only redundant or uninformative tasks without shrinking the corpus, thus increasing task diversity and quality.

Table 1: Dataset-level selection and column-level pre-processing ablations. Results are **IQM** with 95% CI; AUC/accuracy on CC-18, correlation/R^2 on CTR-23, Avg. is unweighted. First row is the full-corpus baseline; dataset-level rows test deduplication, ≥ 30 feature filtering, and both. Column-level **Setup** = (NaN, Corr, mf-5): '+' enabled, '–' disabled; NaN drops columns with $> 50\%$ missing, Corr is Pearson (P) or Spearman (S) deduplication at threshold 0.90, and mf-5 skips datasets with < 5 surviving features. Best column-level IQM per metric is **bold**.

Setup	AUC	Acc.	Corr.	R^2	Avg.
Baseline	$0.9757_{[.9731,.9776]}$	$0.9233_{[.9199,.9265]}$	$0.9021_{[.8993,.9047]}$	$0.8145_{[.8097,.8188]}$	0.9039
Dataset-level selection					
Deduped	$0.9752_{[.9728,.9773]}$	$0.9236_{[.9202,.9269]}$	$0.9019_{[.8989,.9047]}$	$0.8146_{[.8101,.8189]}$	0.9038
≥ 30	$0.9728_{[.9704,.9748]}$	$0.9208_{[.9175,.9241]}$	$0.8882_{[.8832,.8930]}$	$0.7890_{[.7812,.7966]}$	0.8927
≥ 30 + dedup	$0.9724_{[.9701,.9744]}$	$0.9195_{[.9159,.9231]}$	$0.8871_{[.8817,.8921]}$	$0.7870_{[.7788,.7949]}$	0.8915
Column-level pre-processing					
(+, –, –)	$0.9760_{[.9734,.9779]}$	$0.9256_{[.9226,.9286]}$	$0.9078_{[.9049,.9105]}$	$0.8221_{[.8177,.8261]}$	0.9079
(–, P, –)	$0.9760_{[.9737,.9780]}$	$0.9251_{[.9222,.9280]}$	$0.9043_{[.9019,.9065]}$	$0.8195_{[.8157,.8230]}$	0.9062
(–, S, –)	$0.9758_{[.9734,.9778]}$	$0.9238_{[.9209,.9267]}$	$0.9032_{[.9006,.9055]}$	$0.8164_{[.8124,.8201]}$	0.9048
(+, –, +)	$\mathbf{0.9763}_{[.9737,.9783]}$	$0.9251_{[.9221,.9280]}$	$0.9079_{[.9053,.9103]}$	$0.8249_{[.8207,.8289]}$	0.9085
(+, P, –)	$0.9756_{[.9732,.9777]}$	$0.9239_{[.9204,.9272]}$	$0.9027_{[.9002,.9050]}$	$0.8166_{[.8121,.8205]}$	0.9047
(–, P, +)	$0.9759_{[.9734,.9778]}$	$0.9250_{[.9220,.9280]}$	$\mathbf{0.9098}_{[.9071,.9125]}$	$\mathbf{0.8281}_{[.8236,.8323]}$	**0.9097**
(+, P, +)	$0.9756_{[.9731,.9775]}$	$0.9248_{[.9219,.9277]}$	$0.9060_{[.9034,.9083]}$	$0.8227_{[.8186,.8264]}$	0.9073
(+, S, –)	$0.9758_{[.9734,.9778]}$	$0.9241_{[.9215,.9267]}$	$0.9057_{[.9031,.9082]}$	$0.8203_{[.8161,.8241]}$	0.9065
(–, S, +)	$0.9762_{[.9737,.9782]}$	$0.9255_{[.9228,.9282]}$	$0.9078_{[.9053,.9102]}$	$0.8241_{[.8203,.8278]}$	0.9084
(+, S, +)	$\mathbf{0.9763}_{[.9738,.9784]}$	$\mathbf{0.9272}_{[.9241,.9302]}$	$0.9077_{[.9055,.9097]}$	$0.8251_{[.8213,.8284]}$	0.9091

3.3 Does this extend to larger models, context windows, and other evaluation benchmarks?

We next ask whether the gains from column-level pre-processing generalize beyond the base model size, training context, and evaluation setting. To this end, we scale up along two axes simultaneously: model capacity (depth and width) and training context length. We train a deeper, wider model under three maximum training context lengths: 1,024 ("1k", base), 2,048 ("2k"), and 4,096 ("4k") tokens. Each variant is trained with and without pre-processing (we choose NaN + Spearman + mf-5, denoted "+pp"). Inference continues to use context size 2,048 and no pre-processing for consistency. Full architectural, training, and context size details for all six variants are listed in Appendix B.2.

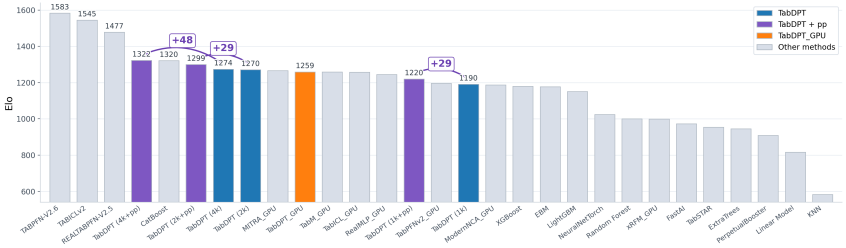


Figure 5: Pre-processing yields Elo gains at all context sizes. Purple bars are models trained with pre-processing (+pp), blue bars their corresponding baselines, and grey bars other TabArena methods. TabDPT [22] is in orange.

We evaluate on TabArena's 51 tabular prediction tasks (Apache License 2.0) [8]. We compute pairwise win rates across all configurations. The results are directionally consistent with our findings: the pre-processed variant outperforms its baseline counterpart at every context scale: 4k+pp beats 4k on 67% of tasks, 2k+pp beats 2k on 61%, and 1k+pp beats 1k on 63%. The strongest overall configuration is 4k+pp. We contextualize these findings within the broader TabArena leaderboard in Figure 5, reporting Elo against some other default-configuration baselines, with details in Appendix C.

4 Understanding what makes generalization possible

The previous sections established two key empirical facts: first, that broad transfer can emerge from pre-training on a single table, and second, that this behaviour persists and can be systematically shaped when scaling to large corpora. These observations cannot be easily explained as parametric transfer learning where predictive signals from one dataset transfer to a related one. Instead, the model must have learned a general prediction algorithm, and we argue that tabular ICL models specifically rely on a *learned retrieval-and-aggregation procedure*. In that sense, their form of generalization is closer to that of k-nearest neighbours than a standard fitted predictor—indeed, it would not be surprising for a non-parametric method such as kNN to perform reasonably well on both an MNIST-like task and a CALIFORNIA-HOUSING-like task. The role of pre-training is thus not to store a universal decision rule in the weights, but rather to learn how to compare examples, identify context points relevant to a query, and aggregate their label information.

This retrieval-based perspective stands in contrast to previous works on prior-fitted networks, which interpret TFM pre-training as inducing a prior that corresponds to the space of processes that generate pre-training datasets [16, 24]. In Section 2, however, we observed that TFMs are effective on data outside of their training distribution, transferring between entirely different tabular datasets, and furthermore are effective when pre-trained with a very limited distribution. The prior-fitting account still does not explain the case where pre-training and inference distributions have no overlap, which motivates our alternative retrieval-and-aggregation perspective of model behaviour. Concurrent work casts further doubt on the prior-fitting account by showing that the predictions of current TFMs are neither marginalization- nor factorization-consistent, and therefore cannot be the conditionals of any single joint distribution over a table's columns [19]. We concede that approximate Bayesian inference could be occurring in models trained with broader distributions, but we believe that learned retrieval is a sufficient explanation for these models' capabilities, while being both more parsimonious and consistent with our observations.

In this section, we provide evidence for learned retrieval through three controlled analyses. First, we show that models that generalize well are also better at retrieving exact label information from the context. Second, we show that forcing the model to behave in a more explicit similarity-based way does not hurt performance, and can even help in weaker regimes. Finally, we show that strong models tend to converge toward similar retrieval patterns, suggesting that there exists a fairly universal structure underlying good tabular generalization.

4.1 Generalization is tied to the ability to retrieve from the context

Based on our interpretation, the ability to generalize to new datasets should be tightly linked to a more primitive capability: the ability to identify and use the relevant information already present in the context. A model that cannot reliably retrieve useful in-context examples would have little hope of producing robust predictions on novel datasets. Conversely, a model that does generalize well should also perform well on tasks where the solution is effectively already available in the context and the main challenge is to recover it. To test this, we construct a controlled retrieval task. For each model pre-trained on a single table, for evaluation, we run standard ICL inference, but **with the query rows set to be the same as context rows**, effectively testing whether the model can correctly retrieve and match identical inputs. Concretely, we use the CC-18 classification datasets with a stratified subsample of up to 1024 examples. The model is then given a sequence consisting of labelled examples followed by the same examples without labels. Performance is evaluated using AUC against the true labels, which we call **retrieval AUC**.

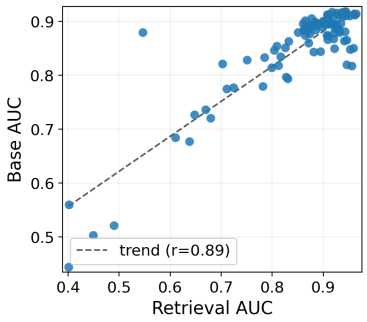


Figure 6: In-context retrieval AUC (x-axis) vs. base AUC (y-axis) for 88 models each pre-trained on one dataset. Each point is one model.

Because every query row has an exact copy with its label in the context, a model that reliably attends to and copies the matching support label should score highly, although not perfectly: aggregation over several context rows can dilute or override the exact match, exactly as in kNN with $k > 1$. A model that ignores the context and relies purely on parametric knowledge would revert to marginal

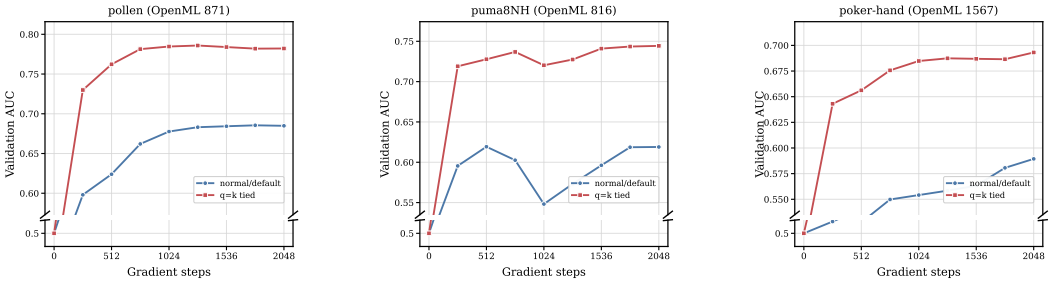
or random predictions. The procedure is repeated with $n = 3$ independent stratified subsamples per dataset and the scores are averaged. We report the mean score across CC-18 datasets. The reported **base AUC** for each model is its held-out test-set AUC on CC-18 (standard ICL with distinct query and context rows) which we use as a proxy for model performance. This highlights the *retrieve and copy* skill as a core mechanism that tabular ICL must learn in order to generalize, similarly to the induction head mechanism in LLMs [25].

Results Figure 6 shows results for 88 single-table pre-trained models. We find a strong correlation (Pearson = 0.89) between the standard generalization performance on held-out subsets and models' performance in this retrieval-oriented setting. Models that are weak at out-of-distribution generalization are also weak at exploiting the labelled context itself, whereas models that generalize well are precisely those that are able to recover the relevant support information. This supports the view that tabular ICL is not primarily driven by memorizing a broad family of parametric predictors in the weights. Instead, a substantial part of the problem appears to consist of learning a robust similarity-based procedure for retrieving and aggregating information from the context. We stress that exact-copy retrieval and downstream performance are distinct capabilities, measured under different prediction protocols, so the two AUCs are not comparable in absolute terms. The informative signal is how they vary together.

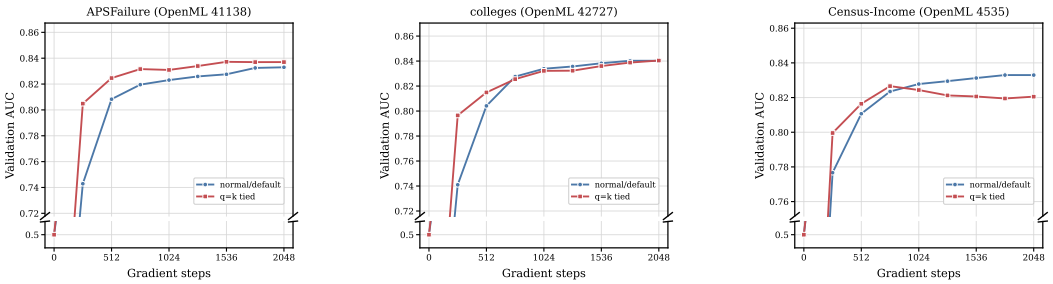
4.2 Forcing a retrieval-based attention mechanism improves performance for weaker models

If good generalization relies on learned retrieval, then one should expect tabular ICL models to be relatively robust to interventions that make them behave more explicitly like nearest-neighbour methods. Specifically, if such models identify similar context points and aggregate their labels, then constraining the attention mechanism to a simpler similarity rule should not degrade performance, but actually improve it in regimes where the model fails to generalize, by providing a useful inductive structure.

We evaluate this by sharing the query and key projection weights for each attention layer, thus making each layer more "kNN-like" as the attention similarity score explicitly encodes a weighted inner product. Indeed, as we use *qk*-normalization, the attention score between points x_i and x_j when setting $W_Q = W_K$ becomes $q_i^\top q_j = \text{constant} - \frac{1}{2}\|q_i - q_j\|^2$. This intervention strictly reduces the expressiveness of the attention mechanism—making it closer to a Gaussian kNN—and tests whether the model would only require a symmetric comparison operator in order to generalize.



(a) Tables that generalize poorly: `pollen`, `puma8NH`, `poker-hand` (left to right).



(b) Tables that already generalize well: `APSFailure`, `Colleges`, `Census-Income` (left to right).

Figure 7: Ablation forcing $W_Q = W_K$ on six single-table pre-training corpora, grouped by how well the default architecture generalizes with single-table pre-training.

We first target pre-training tables that generalize *poorly*, *i.e.*, tables whose resulting models attain low downstream AUC across the evaluation suite (selected using Table 3). This is where (i) the default architecture struggles to exploit labelled context examples from the pre-training table alone, and (ii) where the $W_Q = W_K$ constraint should help most under the retrieval interpretation. In Figure 7a, the simplified version wins on all three such tables (`pollen`, `puma8NH`, `poker-hand`), by $+0.1088$ AUC on average, with analogous results on two further weak tables in Appendix E. On three tables that already generalize well (`APSFailure`, `Colleges`, `Census-Income`), where an imposed similarity rule has little left to contribute, the constraint gives no systematic gain but also no substantial degradation, with a mean difference of -0.0027 AUC (Figure 7b). Sharing the query and key projections therefore supplies a retrieval-inducing bias that matters when the pre-training table is too weak to teach a stable comparison rule, and becomes redundant, or at most mildly restrictive, once it is not. Finally, we scale the intervention to the full OpenML pre-training corpus used in Section 3, where the two architectures are effectively tied: the constraint changes AUC by $+0.0001$ (Figure 8). We do not claim that $W_Q = W_K$ is a better architecture, as this gap is negligible. The relevant point is that reducing attention to a strictly symmetric similarity rule has no noticeable cost even given abundant and diverse pre-training data, given the model sizes we used in our experiments. Along with the weak-table gains, this reinforces the interpretation of tabular ICL as mostly retrieval-based in this setting, since distances between tokens under $W_Q = W_K$ are symmetric and analogous to soft-kNN.

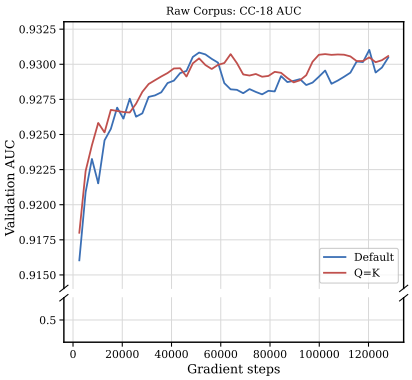


Figure 8: Mean CC-18 AUC for the default architecture and the $W_Q = W_K$ variant, both pre-trained on the full OpenML corpus of Section 3.

4.3 Strong models converge toward similar retrieval patterns

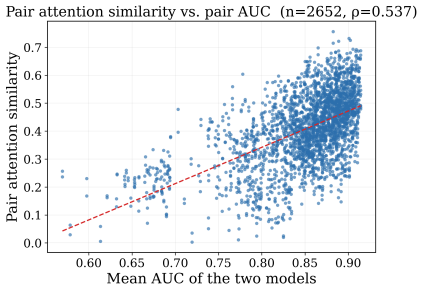


Figure 9: Pair attention similarity vs. mean AUC, $n = 2652$ points (one per model pair $\times$ probe dataset), averaged over blocks. Spearman correlation is 0.537. Per-block results are in Figure 11.

We further show that successful models should not merely perform well, but should do so in similar ways. If there exists a fairly universal retrieval strategy that works across tabular problems, then independently trained strong models should tend to converge toward comparable attention patterns. By contrast, if performance arose from brittle or highly idiosyncratic solutions, one would expect much more diversity across models.

We test this prediction directly on 52 TabDPT checkpoints each trained on a single OpenML dataset (single attention head per block for simplicity, model and training details are in Appendix D). We probe each model on two further held-out OpenML datasets via forward passes to extract attention maps. For each (model, probe), we record the per-block, per-query attention distribution at a fixed in-context cutoff: the softmax-normalized vector a query token assigns over the C context tokens. We compare two models row-by-row using Spearman rank correlation, averaging over queries and probe examples to obtain one similarity scalar per (model pair, block, probe) and then average the resulting scalar over the 12 blocks, yielding one pair-level value per (model pair, probe). Across them, the Spearman correlation between mean held-out AUC and pair attention similarity is 0.537 (Figure 9).

To isolate what drives this correlation, we group models by performance, splitting by AUC quartile. We form three pair categories: strong–strong (SS, both endpoints in the top quartile), strong–weak (SW), and weak–weak (WW, both in the bottom quartile). The results of this experiment are in Table 2 and the

Table 2: Pair attention similarity by category. CIs are non-parametric bootstrap intervals (10,000 resamples) on the difference of group means.

Group	Mean	n	vs.	Δ	95% CI
SS	**0.504**	156	SW	$+0.163$	$[+0.143, +0.183]$
SW	0.341	338	WW	$+0.051$	$[+0.027, +0.075]$
WW	0.290	156			

full methodological details (model specification, similarity definition, bootstrap-CI procedure, etc.) are in Appendix D, which also contains per-block correlation values.

This convergence is unlikely to be accidental. Although checkpoints are trained on different datasets, high-AUC models independently discover similar attention patterns, as expected if good generalization follows a common retrieval principle. Together with the preceding analyses, these results suggest that tabular ICL models generalize not because they encode one gigantic universal predictor in their weights, but because they learn how to use the context at inference time. A simple similarity rule is often already enough to obtain non-trivial transfer, which explains why generalization is possible at all. The strongest models then improve upon this baseline by learning a more refined, more robust notion of relevance and aggregation.

5 Related Work

Although deep learning and gradient boosted tree ensemble algorithms are currently dominant on tabular benchmarks [8], our work demonstrates a connection to k-nearest neighbours [10] and possibly kernel-based extensions [36]. Explicit k-nearest retrieval has had recent applications as part of tabular ICL models [32, 22, 34].

While previous works on tabular ICL have explored many aspects of training task creation, they have generally focused on scaling the diversity of tasks, either through complex synthetic priors [16, 17, 27, 28, 34, 35] or augmented real world data [22, 12], and the regime of extremely limited pre-training data has not been explored. Other surprising generalization capabilities of tabular ICL models under modified training approaches include resilience to distribution shifts [15] and causal inference [2, 30].

Substantial literature exists analyzing and extending in-context learning in large language models [7], however this is an emergent capability seen in autoregressive next-token prediction models trained on large amounts of text data, which is a fundamentally different setting from TFMs that directly train on an ICL objective. The observations in this paper are therefore separate from LLM ICL behaviour. While some LLM-focused papers have evaluated non-text data (e.g., [5, 29, 13]), they have still focused on autoregressive sequence modelling in ways that make them inapplicable to the models discussed here. Tabula-8B [11] is an exception among tabular ICL models in that it uses a language model backbone, so we exclude it from our scope, but LLM-focused ICL findings could apply to it.

6 Conclusion and limitations

In this paper we investigated how pre-training data enables out-of-distribution generalization in TFMs. Taking a task-centric view, we explained why strong generalization can emerge even when training on a single real table, and expanded on this finding to make practical recommendations regarding dataset-level and column-level data filtering. Finally, we argued for a retrieval-focused interpretation of TFMs rather than one based on standard transfer learning or Bayesian inference, providing empirical evidence for retrieval behaviours. Overall, our findings challenge existing assumptions about how TFMs work and what actually matters when scaling TFM pre-training data.

Although we analyzed TFM behaviour across a broad range of datasets, model sizes, and context lengths, we acknowledge other limitations in our work. We were unable to definitively identify or exclude additional mechanisms beyond learned retrieval that may underlie out-of-distribution generalization. Further research is therefore required to achieve a more comprehensive understanding of how in-context learning operates in TFMs. Moreover, our investigation was restricted to pre-training data derived from real-world datasets, and consequently, our findings may not fully extend to scenarios in which models are trained exclusively on synthetic data. Finally, while transformer architectures are universal across modern TFMs, they are split between row-based and cell-based attention, and our results only used row-based attention following TabDPT and TabPFNv1.

7 Acknowledgements

This work was supported by Mitacs through the Mitacs Accelerate program, and was enabled in part by compute resources provided by Mila (mila.quebec) and the Digital Research Alliance of Canada. We also thank Rasa Hosseinzadeh for the valuable discussions and ideas that contributed to this work.

References

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

[2] Vahid Balazadeh Meresht, Hamidreza Kamkari, Valentin Thomas, Junwei Ma, Bingru Li, Jesse Cresswell, and Rahul Krishnan. CausalPFN: Amortized causal effect estimation via in-context learning. In *Advances in Neural Information Processing Systems*, 2025.

[3] Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan van Rijn, and Joaquin Vanschoren. OpenML benchmarking suites. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[4] Bernd Bischl, Giuseppe Casalicchio, Taniya Das, Matthias Feurer, Sebastian Fischer, Pieter Gijsbers, Subhaditya Mukherjee, Andreas C Müller, László Németh, Luis Oala, Lennart Purucker, Sahithya Ravi, Jan N van Rijn, Prabhant Singh, Joaquin Vanschoren, Jos van der Velde, and Marcel Wever. OpenML: Insights from 10 years and more than a thousand papers. *Patterns*, 6 (7), 2025.

[5] Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, 2022.

[6] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SigKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[7] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, 2024.

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

[9] Sebastian Felix Fischer, Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – A curated tabular regression benchmarking suite. In *AutoML Conference (Workshop)*, 2023.

[10] Evelyn Fix and J. L. Hodges. *Discriminatory Analysis: Nonparametric Discrimination, Consistency Properties*, volume 1. USAF School of Aviation Medicine, 1951.

[11] Josh Gardner, Juan C Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling. In *Advances in Neural Information Processing Systems*, 2024.

[12] Anurag Garg, Muhammad Ali, Noah Hollmann, Lennart Purucker, Samuel Müller, and Frank Hutter. Real-TabPFN: Improving tabular foundation models via continued pre-training with real-world data. *arXiv:2507.03971*, 2025.

[13] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, 2022.

[14] Pieter Gijsbers. Colleges. OpenML Dataset 42727, 2020. URL `https://www.openml.org/d/42727`. Accessed: 2025-10-21.

[15] Kai Helli, David Schnurr, Noah Hollmann, Samuel Müller, and Frank Hutter. Drift-resilient TabPFN: In-context learning temporal distribution shifts on tabular data. In *Advances in Neural Information Processing Systems*, 2024.

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

[17] 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.

[18] Rasa Hosseinzadeh, Alex Labach, Zexin Xue, Shuyi Han, Valentin Thomas, and Anthony L Caterini. TabDPT-Turbo: Efficient in-context learning for tabular prediction. *arXiv:2608.01400*, 2026.

[19] Christian Klötergens, Vijaya Krishna Yalavarthi, Lars Schmidt-Thieme, and Tom Hanika. Do tabular foundation models agree with themselves? *arXiv:2608.06004*, 2026.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, 2022.

[22] Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems*, 2025.

[23] 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*, 2023.

[24] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022.

[25] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *arXiv:2209.11895*, 2022.

[26] R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

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

[28] 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*, 2026.

[29] Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. *arXiv:2312.03002*, 2023.

[30] Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-PFN: In-context learning for causal effect estimation. In *Advances in Neural Information Processing Systems*, 2025.

[31] Marco Spinaci, Marek Polewczyk, Maximilian Schambach, and Sam Thelin. ConTextTab: A semantics-aware tabular in-context learner. In *Advances in Neural Information Processing Systems*, 2025.

[32] Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L Caterini. Retrieval & fine-tuning for in-context tabular models. *Advances in Neural Information Processing Systems*, 2024.

[33] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

[34] 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:2509.03505*, 2025.

[35] Xiyuan Zhang, Danielle Maddix Robinson, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael Mahoney, Tony Hu, Huzefa Rangwala, George Karypis, and Yuyang (Bernie) Wang. Mitra: Mixed synthetic priors for enhancing tabular foundation models. In *Advances in Neural Information Processing Systems*, 2025.

[36] Wangmeng Zuo, David Zhang, and Kuanquan Wang. On kernel difference-weighted k-nearest neighbor classification. *Pattern Analysis and Applications*, 11(3):247–257, 2008.

A Impact of pre-training dataset on CC-18 and CTR-23 Performance

Tables 3 and 4 report the performance of TabDPT when pre-trained on different datasets and evaluated on two benchmark suites: **CC-18** for classification (measured by AUC) and **CTR-23** for regression (measured by R^2). Each table lists the results for all pre-training datasets, along with their number of instances, their number of features, and their domain. The results are sorted in descending order of performance, highlighting which pre-training datasets achieve better performance on each benchmark.

Table 3: AUC scores on CC-18 for TabDPT pre-trained on various datasets, sorted in descending order of AUC.

Dataset	AUC on CC-18	# Instances	# Features	Dataset Domain
APSFailure	0.918	76000	171	industrial-operational
colleges	0.918	7063	45	other
ipums_la_99-small	0.917	8844	57	financial-demographic
ipums_la_97-small	0.916	7019	61	financial-demographic
ipums_la_98-small	0.915	7485	56	financial-demographic
dionis	0.915	416188	61	other
cardiotocography	0.914	2126	36	medical-human-sensor
Census-Income	0.914	299285	42	financial-demographic
volkert	0.912	58310	181	other
KDDCup09_appetency	0.912	50000	231	human-behaviour
covertype	0.912	581012	55	biology-ecology
gas-drift-different-concentrations	0.911	13910	130	other-science
KDDCup09_churn	0.911	50000	231	industrial-operational
road-safety	0.910	111762	33	human-behaviour
kick	0.910	72983	33	industrial-operational
eye_movements	0.908	10936	28	medical-human-sensor
gas-drift	0.906	13910	129	other-science
philippine	0.905	5832	309	other
dilbert	0.904	10000	2001	other
christine	0.901	5418	1637	other
MiniBooNE	0.900	130064	51	physics-astronomy
Satellite	0.900	5100	37	physics-astronomy
jasmine	0.899	2984	145	other
scene	0.899	2407	300	vision-audio-text
helena	0.896	65196	28	other
kdd_internet_usage	0.896	10108	69	financial-demographic
SpeedDating	0.895	8378	121	human-behaviour
pol	0.895	10082	27	industrial-operational
elevators	0.894	16599	19	other
musk	0.894	6598	168	other-science
one-hundred-plants-texture	0.894	1599	65	biology-ecology
heloc	0.894	10000	23	financial-demographic
default-of-credit-card-clients	0.893	13272	21	financial-demographic
ada_agnostic	0.892	4562	49	financial-demographic
sylva_agnostic	0.889	14395	217	biology-ecology
colleges_usnews	0.889	1302	34	other
spoken-arabic-digit	0.887	263256	15	vision-audio-text
ada	0.887	4147	49	other
jannis	0.885	83733	55	other
PizzaCutter3	0.885	1043	38	other
fbis.wc	0.884	2463	2001	vision-audio-text
porto-seguro	0.883	595212	58	human-behaviour
sylvine	0.883	5124	21	other
cjs	0.880	2796	35	biology-ecology
Click_prediction_small	0.879	39948	12	human-behaviour
guillermo	0.879	20000	4297	other

Table 3: AUC scores on CC-18 for TabDPT pre-trained on various datasets, sorted in descending order of AUC.

Dataset	AUC on CC-18	# Instances	# Features	Dataset Domain
mushroom	0.879	8124	23	biology-ecology
PieChart3	0.877	1077	38	other
higgs	0.877	98050	29	physics-astronomy
magic	0.874	19020	11	physics-astronomy
ada_prior	0.874	4562	15	financial-demographic
MagicTelescope	0.873	19020	12	physics-astronomy
eeg-eye-state	0.867	14980	15	medical-human-sensor
JapaneseVowels	0.866	9961	15	vision-audio-text
shuttle	0.865	58000	10	physics-astronomy
ldpa	0.863	164860	8	medical-human-sensor
okcupid-stem	0.863	50789	20	human-behaviour
house_16H	0.860	22784	17	financial-demographic
credit	0.854	16714	11	financial-demographic
house_8L	0.851	22784	9	financial-demographic
artificial-characters	0.850	10218	8	deterministic-simulated
page-blocks	0.849	5473	11	vision-audio-text
sf-police-incidents	0.847	2215023	9	human-behaviour
albert	0.846	425240	79	other
walking-activity	0.844	149332	5	medical-human-sensor
pbcseq	0.843	1945	19	medical-human-sensor
fabert	0.834	8237	801	other
compas-two-years	0.833	4966	12	human-behaviour
Diabetes130US	0.828	71090	8	medical-human-sensor
madeline	0.821	3140	260	other
airlines	0.820	539383	8	industrial-operational
analcatdata_halloffame	0.818	1340	17	other
visualizing_soil	0.817	8641	5	biology-ecology
analcatdata_supreme	0.814	4052	8	other
chess	0.796	28056	7	deterministic-simulated
kropt	0.794	28056	7	deterministic-simulated
kr-vs-k	0.780	28056	7	deterministic-simulated
poker-hand	0.777	1025009	11	deterministic-simulated
pollen	0.775	3848	6	biology-ecology
puma8NH	0.736	8192	9	deterministic-simulated
yeast	0.727	1269	9	biology-ecology
hill-valley	0.720	1212	101	deterministic-simulated
imdb.drama	0.684	120919	1002	vision-audio-text
twonorm	0.677	7400	21	deterministic-simulated
nursery	0.560	12958	9	human-behaviour
gametes_epistasis	0.521	1600	21	deterministic-simulated
mofn-3-7-10	0.504	1324	11	deterministic-simulated
parity5_plus_5	0.444	1124	11	deterministic-simulated

Table 4: R^2 scores on CTR-23 for TabDPT pre-trained on various datasets, sorted in descending order of R^2.

Dataset	R^2 on CTR-23	# Instances	# Features	Dataset Domain
colleges	0.693	7063	45	other
kick	0.693	72983	33	industrial-operational
ipums_la_97-small	0.691	7019	61	financial-demographic
dionis	0.685	416188	61	other
ipums_la_98-small	0.685	7485	56	financial-demographic

Table 4: R^2 scores on CTR-23 for TabDPT pre-trained on various datasets, sorted in descending order of R^2.

Dataset	R^2 on CTR-23	# Instances	# Features	Dataset Domain
Census-Income	0.684	299285	42	financial-demographic
APSFailure	0.682	76000	171	industrial-operational
ipums_la_99-small	0.681	8844	57	financial-demographic
KDDCup09_churn	0.679	50000	231	industrial-operational
covertype	0.679	581012	55	biology-ecology
road-safety	0.678	111762	33	human-behaviour
gas-drift-different-concentrations	0.677	13910	130	other-science
gas-drift	0.672	13910	129	other-science
eye_movements	0.672	10936	28	medical-human-sensor
musk	0.671	6598	168	other-science
KDDCup09_appetency	0.671	50000	231	human-behaviour
volkert	0.669	58310	181	other
MiniBooNE	0.668	130064	51	physics-astronomy
sylvine	0.667	5124	21	other
cardiotocography	0.665	2126	36	medical-human-sensor
dilbert	0.665	10000	2001	other
default-of-credit-card-clients	0.664	13272	21	financial-demographic
sylva_agnostic	0.657	14395	217	biology-ecology
helena	0.655	65196	28	other
jannis	0.649	83733	55	other
pol	0.647	10082	27	industrial-operational
scene	0.647	2407	300	vision-audio-text
philippine	0.646	5832	309	other
heloc	0.645	10000	23	financial-demographic
one-hundred-plants-texture	0.643	1599	65	biology-ecology
magic	0.643	19020	11	physics-astronomy
elevators	0.641	16599	19	other
MagicTelescope	0.640	19020	12	physics-astronomy
house_16H	0.639	22784	17	financial-demographic
PieChart3	0.638	1077	38	other
ldpa	0.635	164860	8	medical-human-sensor
JapaneseVowels	0.634	9961	15	vision-audio-text
page-blocks	0.633	5473	11	vision-audio-text
christine	0.633	5418	1637	other
spoken-arabic-digit	0.631	263256	15	vision-audio-text
higgs	0.630	98050	29	physics-astronomy
eeg-eye-state	0.630	14980	15	medical-human-sensor
colleges_usnews	0.629	1302	34	other
jasmine	0.627	2984	145	other
SpeedDating	0.627	8378	121	human-behaviour
Satellite	0.623	5100	37	physics-astronomy
cjs	0.621	2796	35	biology-ecology
porto-seguro	0.620	595212	58	human-behaviour
PizzaCutter3	0.617	1043	38	other
Click_prediction_small	0.615	39948	12	human-behaviour
ada_prior	0.612	4562	15	financial-demographic
shuttle	0.608	58000	10	physics-astronomy
sf-police-incidents	0.607	2215023	9	human-behaviour
artificial-characters	0.604	10218	8	deterministic-simulated
pbcseq	0.585	1945	19	medical-human-sensor
airlines	0.585	539383	8	industrial-operational
house_8L	0.581	22784	9	financial-demographic
ada	0.581	4147	49	other
kdd_internet_usage	0.579	10108	69	financial-demographic

Table 4: R^2 scores on CTR-23 for TabDPT pre-trained on various datasets, sorted in descending order of R^2.

Dataset	R^2 on CTR-23	# Instances	# Features	Dataset Domain
kropt	0.578	28056	7	deterministic-simulated
analcatdata_halloffame	0.575	1340	17	other
albert	0.575	425240	79	other
chess	0.574	28056	7	deterministic-simulated
ada_agnostic	0.571	4562	49	financial-demographic
okcupid-stem	0.567	50789	20	human-behaviour
guillermo	0.565	20000	4297	other
kr-vs-k	0.560	28056	7	deterministic-simulated
walking-activity	0.558	149332	5	medical-human-sensor
credit	0.558	16714	11	financial-demographic
compas-two-years	0.557	4966	12	human-behaviour
visualizing_soil	0.545	8641	5	biology-ecology
fabert	0.544	8237	801	other
analcatdata_supreme	0.543	4052	8	other
poker-hand	0.512	1025009	11	deterministic-simulated
Diabetes130US	0.500	71090	8	medical-human-sensor
fbis.wc	0.483	2463	2001	vision-audio-text
madeline	0.478	3140	260	other
pollen	0.424	3848	6	biology-ecology
puma8NH	0.367	8192	9	deterministic-simulated
twonorm	0.316	7400	21	deterministic-simulated
yeast	0.278	1269	9	biology-ecology
hill-valley	0.221	1212	101	deterministic-simulated
mofn-3-7-10	0.131	1324	11	deterministic-simulated
mushroom	0.112	8124	23	biology-ecology
parity5_plus_5	-0.048	1124	11	deterministic-simulated
nursery	-0.228	12958	9	human-behaviour
gametes_epistasis	-0.806	1600	21	deterministic-simulated
imdb.drama	-16.712	120919	1002	vision-audio-text

B Training and architecture details

Throughout this paper, "TabDPT" refers specifically to TabDPT v1.1.

B.1 Base model: many-dataset pre-training ablations

This section provides the experimental setup for the many-dataset pre-training ablations. The deduplication, dataset-level filtering, and column-level pre-processing experiments all use the same base TabDPT architecture and training configuration described henceforth.

Pre-training corpus We use part of the datasets collected by Hosseinzadeh et al. [18] as our pre-training corpus. This contains $1{,}732$ datasets collected from OpenML [4], with datasets contained in the CC-18, CTR-23, or TabArena [8] benchmarks automatically removed. We do not filter datasets with fewer than 10 columns, unlike the cited paper, but we do exclude datasets with no clear target variable y. We additionally find that 197 of these datasets are exact duplicates. We cut off each dataset at $10{,}000$ rows and $3{,}000$ columns to bound pre-processing costs.

Architecture We use a 12-layer transformer with embedding dimension $d_{\text{model}} = 512$, 8 attention heads, and MLP hidden dimension 1,024 (hidden factor 2). The model uses post-layer normalization and an asymmetric in-context attention pattern: query (test) tokens attend only to context (train) tokens, never to each other. Feature columns are projected to d_{model} via a shared linear encoder. For context tokens, the target embedding (a separate learned linear layer) is added on top of the feature embedding. At each step, inputs are capped at 100 feature columns and 10 output classes.

Training At each step, one column is selected uniformly at random as the prediction target y while the remaining columns form the feature matrix X. Classification and regression heads are trained jointly in a single forward pass using a self-supervised SSL objective. Label smoothing of 0.1 is applied to the classification cross-entropy loss. We use AdamW with learning rate 5×10^{-4}, weight decay 0.05, and no learning-rate schedule. Gradient norms are clipped to 1.0. Dropout is 0.0 throughout. Each run trains for 128,000 gradient update steps, with per-step batch size of 32 and gradient accumulation over 8 steps for an effective batch size of 256. The context window is randomly sampled at each step and is at minimum 50 tokens and at maximum 1,024.

Data pre-processing and retrieval Each dataset is truncated to at most 10,000 rows and 3,000 columns before any pre-processing. Categorical columns are coerced to integers via label encoding and columns where fewer than 1% of values are interpretable as numeric are treated as categorical. Missing values are imputed with column means. The joint $[X \mid y]$ matrix is then standardized with a `StandardScaler`. A FAISS flat L2 index is built on training-split embeddings and used for retrieval-augmented context assembly during training.

Evaluation We report results on the same two standard benchmarks used previously: **CC-18** for classification and **CTR-23** for regression. On all metrics, we report the **Interquartile Mean** (IQM), which trims the bottom and top quartiles and is robust to outliers. We also report 95% bootstrap confidence intervals computed via stratified bootstrap over datasets using $20{,}000$ resampling iterations. Inference uses context size $2{,}048$ and 8 ensemble passes on the raw benchmark datasets, without applying any column pre-processing, to ensure consistency across experiments.

Hardware Each run uses a single Nvidia A100 (40 GB) GPU with 8 CPU workers and 150GB RAM.

B.2 Large model: context-length and architecture scaling

The experiments in subsection 3.3 use a larger model trained at three context lengths, with and without pre-processing, for six runs total. All six share the configuration described here.

Architecture We increase the model to 16 transformer layers and embedding dimension $d_{\text{model}} = 768$, with 8 attention heads and MLP hidden dimension 1,536 (hidden factor 2). Pre-layer normalization replaces the post-norm used in the base model. All other architectural choices (asymmetric attention, feature/label encoders, output heads, input caps) are unchanged.

Training We use the ScheduleFree AdamW optimizer with learning rate 1×10^{-3} and weight decay 0.05. Training uses BF16 mixed precision. Gradient clipping (1.0) and label smoothing (0.1) are unchanged. Each run trains for 128,000 gradient steps. Batch size is 32 with gradient accumulation over 4 steps, giving effective batch size 128. The three variants differ in the maximum number of labelled context tokens (shown in the main text) but the minimum context size is 100 for all three. Inference for all six runs uses context size $2{,}048$ with no column pre-processing applied, for consistency across comparisons.

Data pre-processing and retrieval Dataset caps, categorical encoding, imputation, scaling, and FAISS L2 retrieval are identical to the base model. For the "+pp" variants, the NaN-column drop, Spearman correlation deduplication (threshold 0.90), and minimum-feature filter ($k = 5$) are applied to each dataset before training, with the column mask fitted on the training split as in the base ablations.

Hardware Each of the six runs uses a single Nvidia H100 GPU. All other software and infrastructure settings are as described in Section B.1.

C Additional TabArena results

We provide extended results on the TabArena benchmark [8]. All experiments use TabArena-Lite (51 tasks, fold 0), bootstrapped 2000 times. Elo is computed against a pool of default-configuration baselines (no tuning or ensembling), and is anchored at Random Forest = 1000. We evaluate six

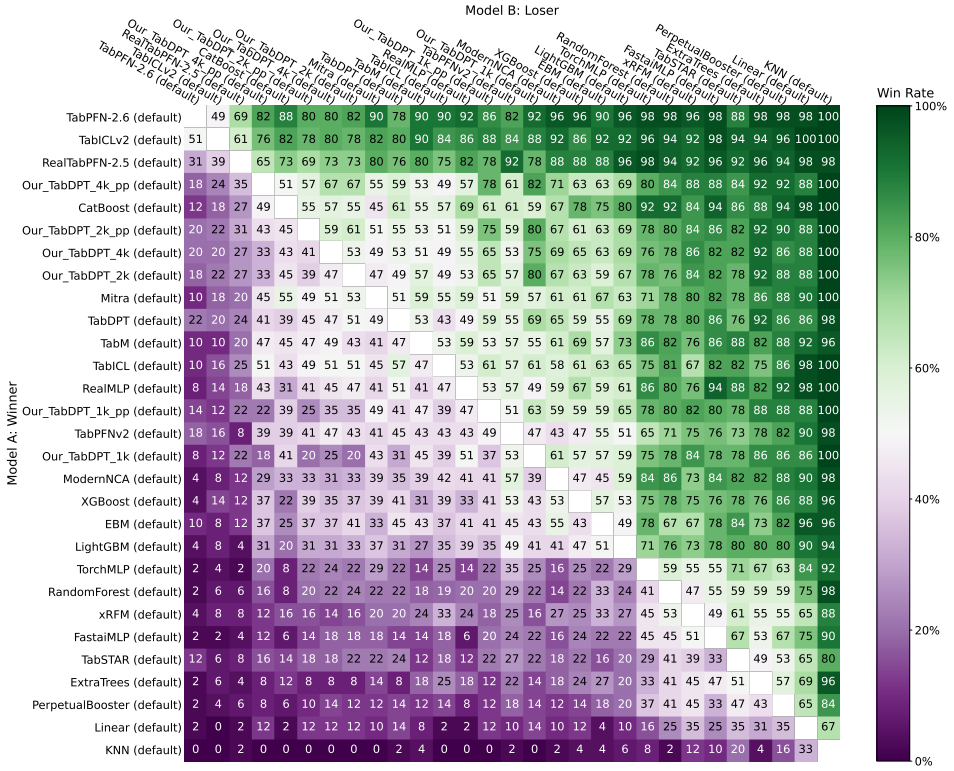


Figure 10: Pairwise win-rate matrix on TabArena-Lite across all evaluated methods (our models and baselines). Each cell shows the fraction of tasks where the row method outperforms or ties the column method.

TabDPT variants spanning a 2×3 grid over pre-processing (+pp vs. none) and training context length (1k, 2k, 4k tokens), alongside a broad set of baseline methods.

C.1 Pairwise win rates

Figure 10 shows pairwise win rates across all methods. The relative ordering among our variants matches the main text: 4k+pp performs best, followed by 2k+pp and 4k, with shorter-context models trailing.

C.2 Leaderboard results

Table 5 reports leaderboard metrics for all evaluated methods. The 4k+pp variant achieves the strongest performance among our models, and the remaining variants follow the same ordering as in Figure 10.

D Attention similarity experiments: methods and supplementary results

D.1 Models, probes, and similarity calculations

Checkpoints We probe 52 TabDPT checkpoints, each pre-trained from scratch on a single OpenML dataset, all sharing the same architecture (12 transformer blocks, $d_{\text{model}} = 512$, single attention head, asymmetric in-context attention pattern) and the same training schedule (lr 5×10^{-4}, weight decay 0.05, sequence length 1536, taken at 12,800 gradient updates with an effective batch size of 256).

Probes Each model is evaluated on two held-out OpenML classification datasets, AMAZON_EMPLOYEE_ACCESS (OpenML ID 4135) and NURSERY (OpenML ID 1568), that none of

Table 5: Full TabArena-Lite leaderboard (51 tasks, fold 0). Elo is anchored at Random Forest = 1000. Our models are in bold.

Method	Elo	Avg. Rank
TABPFN-V2.6	1583	4.16
TABICLv2	1545	4.80
REALTABPFN-V2.5	1477.6	6.16
Our TabDPT (4k+pp)	**1322.8**	**10.25**
CatBoost	1320.8	10.31
Our TabDPT (2k+pp)	**1299.8**	**10.96**
Our TabDPT (4k)	**1274.3**	**11.76**
Our TabDPT (2k)	**1270.7**	**11.88**
MITRA_GPU	1265.8	12.04
TabDPT_GPU	1259.1	12.25
TabM_GPU	1258.4	12.27
TabICL_GPU	1257.2	12.31
RealMLP_GPU	1245.1	12.71
Our TabDPT (1k+pp)	**1220.0**	**13.53**
TabPFNv2_GPU	1195.6	14.33
Our TabDPT (1k)	**1190.8**	**14.49**
ModernNCA_GPU	1187.0	14.62
XGBoost	1179.5	14.86
EBM	1177.1	14.94
LightGBM	1151.5	15.78
NeuralNetTorch	1024.4	19.75
Random Forest (anchor)	1000.0	20.44
xRFM_GPU	999.6	20.45
FastAI	973.1	21.18
TabSTAR	954.4	21.67
ExtraTrees	945.9	21.88
PerpetualBooster	908.8	22.78
Linear Model	816.1	24.71
KNN	583.2	27.71

the 52 models was trained on. For each (model, probe) we record the per-block attention tensor of shape $(B, H, Q, C) = (32, 1, 512, 512)$ at a fixed in-context cutoff $\mathrm{eval_pos} = 512$, where $\alpha_{b,h,q,c} = \mathrm{softmax}_c(\langle \mathbf{q}_{b,h,q}, \mathbf{k}_{b,h,c} \rangle / \sqrt{d_{\text{head}}})$ is a probability vector over the C context tokens (the *attention distribution* of query q).

Pair similarity For two models i, j on probe p and block ℓ we compute

$$s_{ij}^{(\ell,p)} = \tfrac{1}{BQ} \sum_{b,q} \rho_{\text{Spearman}}\big(\alpha_{b,q,\cdot}^{(i)},\, \alpha_{b,q,\cdot}^{(j)}\big),$$

which is the row-wise Spearman rank correlation between the two attention distributions, that is then averaged over queries and probe examples. The pair-level scalar in the main paper averages $s_{ij}^{(\ell,p)}$ over the 12 blocks. Spearman is preferred over Pearson because rank correlation is invariant to the exact magnitudes of the attention values.

Held-out AUC Per-model AUC is the AUC on the OpenML CC-18 classification suite at the same training step, which is data disjoint from both the training datasets and the two probes.

D.2 Strong / weak partitioning

We split the 52 models by AUC quartile ($q_{25} = 0.831$, $q_{75} = 0.900$). Models with AUC $\geq q_{75}$ are **strong** (S, 13 models), models with AUC $\leq q_{25}$ are **weak** (W, 13 models). The 26 middle-quartile models are excluded.

Table 6: Models in the strong and weak sets, identified by their training dataset.

Strong (AUC ≥ 0.900)	**Weak (AUC ≤ 0.831)**
APSFAILURE, CENSUS-INCOME, KDDCUP09_APPETENCY, SPEEDDATING, CARDIOTOCOGRAPHY, COLLEGES, COVERTYPE, GAS-DRIFT, IPUMS_LA_98-SMALL, MUSK, POL, ROAD-SAFETY, SCENE	DIABETES130US, AIRLINES, ANALCATDATA_SUPREME, CHESS, HILL-VALLEY, PARITY5_PLUS_5, PBCSEQ, POKER-HAND, POLLEN, PUMA8NH, TWONORM, VISUALIZING_SOIL, YEAST

All 95% CIs in Tables 2 and 7 are obtained by bootstrap. We repeatedly draw a version of each group by sampling its observations with replacement, recompute the gap between the two group means, and collect the resulting gap estimates. The 95% CI is the range from the 2.5th to the 97.5th percentile of that collection. We use $10\,000$ resamples for Table 2 and $2\,000$ for Table 7. The random seed 0 is fixed for reproducibility.

D.3 Per-block breakdown

Table 7 reports, per transformer block, the SS / SW / WW group means together with the SS$-$SW gap and its 95% bootstrap CI ($K = 2\,000$). Each row pools across the two probes, so $n_{\text{SS}} = 156$, $n_{\text{SW}} = 338$, $n_{\text{WW}} = 156$ per block.

Table 7: Per-block group means and SS$-$SW gap with 95% bootstrap CI. One observation per (model pair, probe, layer). Rows whose CI excludes zero are marked †.

Block	SS	SW	WW	SS$-$SW	95% CI
0	0.854	0.696	0.593	$+0.158^{\dagger}$	$[+0.124, +0.193]$
1	0.525	0.442	0.402	$+0.083^{\dagger}$	$[+0.005, +0.164]$
2	0.123	0.047	0.279	$+0.075$	$[-0.027, +0.176]$
3	0.145	0.110	0.156	$+0.035$	$[-0.050, +0.119]$
4	0.551	0.383	0.330	$+0.168^{\dagger}$	$[+0.132, +0.205]$
5	0.584	0.363	0.238	$+0.221^{\dagger}$	$[+0.187, +0.255]$
6	0.577	0.403	0.277	$+0.174^{\dagger}$	$[+0.133, +0.218]$
7	0.303	0.295	0.325	$+0.008$	$[-0.057, +0.075]$
8	0.520	0.368	0.269	$+0.153^{\dagger}$	$[+0.124, +0.181]$
9	0.585	0.363	0.253	$+0.222^{\dagger}$	$[+0.197, +0.249]$
10	0.617	0.213	0.073	$+0.404^{\dagger}$	$[+0.367, +0.439]$
11	0.667	0.407	0.288	$+0.260^{\dagger}$	$[+0.239, +0.284]$

The SS$-$SW gap is positive in all 12 blocks, statistically significant in 9 of them (blocks 0, 1, 4, 5, 6, 8, 9, 10, 11), and largest at block 10.

We also see the global correlation effect shown in Figure 9, being concentrated in the deeper transformer blocks in Figure 11: per-block correlation rises from near zero in early blocks to ≈ 0.6 in blocks 9–11.

D.4 Caveat: pair non-independence

The data points are not fully independent, since each model appears in 51 pairs, so the confidence intervals reported above are somewhat tighter than they would be under a stricter resampling scheme.

E Additional query–key sharing ablation

Figure 12 is an additional analysis of training dynamics when $W_Q = W_K$ is enforced, on two further pre-training tables that generalize poorly, the VISUALIZING SOIL and the CHESS datasets. In both cases, forcing $W_Q = W_K$ yields significantly higher validation AUC than the default architecture,

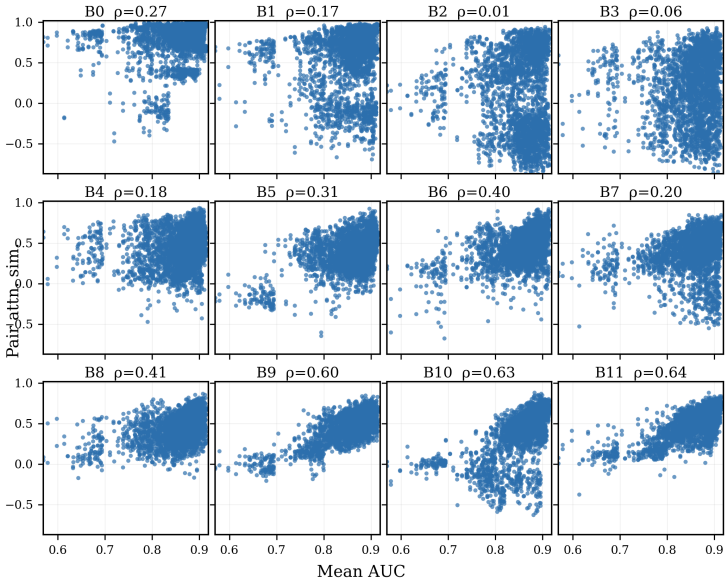


Figure 11: Per-block pair attention similarity vs. mean AUC, $n = 2652$ points (one per model pair $\times$ probe dataset).

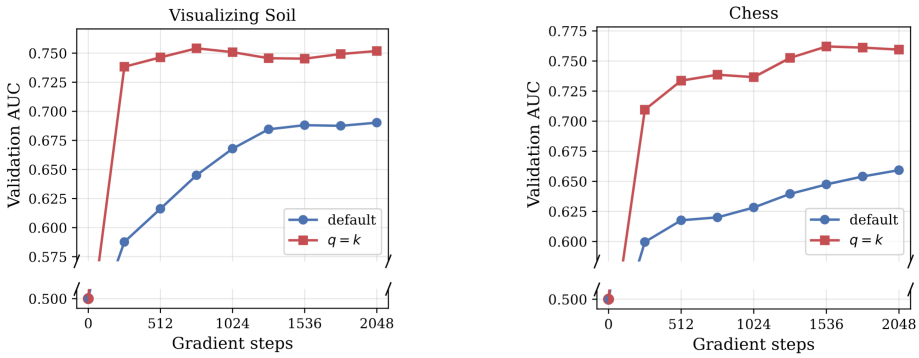


Figure 12: Ablations where $W_Q = W_K$ is enforced, on VISUALIZING SOIL (left) and CHESS (right), using the same setting as Figure 7a.

complementing the main-text weak-table results of Figure 7a and supporting the view that TFM performance is close to a nearest neighbour-like mechanism.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: They accurately summarize the paper's main empirical contributions: single-table pre-training can transfer across domains, task diversity matters more than instance count, column-level pre-processing improves training, and tabular ICL appears retrieval-based. These claims are supported by the experiments in the main text.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present theoretical results or formal proofs. It is purely empirical.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide substantial details to reproduce the core experiments in the appendix, including model architecture, training configuration, pre-processing and retrieval setup, evaluation protocol, bootstrap confidence interval procedure, and hardware.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We use public benchmark datasets and cite the relevant sources, as well as public TabDPT training and inference code, but we do not release our pre-processing and analysis code, model checkpoints, or full reproduction scripts with this submission.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify the experimental settings needed to understand the results, including the pre-training and evaluation datasets, benchmark suites, model architectures, training objectives, optimization settings, context lengths, pre-processing, retrieval, and evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 95% bootstrap confidence intervals for the main IQM benchmark results, bootstrapped TabArena results, and bootstrap confidence intervals for the attention-similarity comparisons, indicating statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For all our experiments, we report the hardware used for the experiments in the appendix, including the GPU type, CPU workers, and memory.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and are not aware of any aspect of the research that violates it. Our work does not involve new human-subject data collection, crowdsourcing, IRB-relevant experiments, or the release of high-risk models or datasets.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Because the paper discusses modelling only in broad, non-specific terms and does not identify concrete connections between tabular modelling and societal harms, we do not believe there are societal impacts that require further discussion.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The work does not introduce high-risk models or datasets requiring special safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use existing public OpenML and TabDPT assets, properly cite them and specify the licenses in the paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not explicitly introduce or release new datasets, models, or code assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology does not involve LLMs as a component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.