TABMF: LEARNING DATASET REPRESENTATIONS WITH TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representations of supervised datasets support higher-level machine-learning tasks such as dataset retrieval, and distribution-shift monitoring. However, existing approaches typically rely either on hand-crafted meta-features or on dataset encoders trained specifically for representation learning.

We introduce TABMF, a framework that repurposes pretrained tabular foundation models (TFMs) as encoders of supervised datasets. TABMF equips a predictive TFM with an architecture-aware dataset-level readout and adapts the model end-to-end using contrastive post-training on stochastic views of synthetic supervised datasets.

We instantiate TABMF on TabICLv2 and TabPFN-2.5 and evaluate the resulting representations across six tasks on unseen real datasets. Simple frozen TFM readouts yield competitive representations, indicating that predictive pretraining already induces substantial dataset-level structure. The two TABMF encoders obtain the strongest aggregate performance among the evaluated learned and classical dataset representations, with particularly strong results in dataset retrieval, compact benchmark construction, and distribution-shift severity assessment. Code is available at `https://anonymous.4open.science/r/TabMF-ICLR`.

1 INTRODUCTION

Many machine-learning problems require reasoning about an entire dataset rather than individual observations. A useful representation of a supervised dataset can support dataset retrieval, benchmark construction, distribution-shift monitoring, transfer of model or hyperparameter knowledge, and identification of rare or atypical datasets. Ideally, such a representation should map a previously unseen dataset to a fixed-dimensional vector whose geometry can be reused across different dataset-level tasks.

Existing approaches largely rely on hand-crafted meta-features or dedicated dataset encoders. Meta-features summarize a dataset through a predefined collection of statistics (Alcobaça et al., 2020) and underpin algorithm selection (McElfresh et al., 2023; Zhao et al., 2021) and zero-shot AutoML (Öztürk et al., 2022; Pineda-Arango et al., 2024), but their representation is fixed by design and cannot adapt to new notions of dataset similarity. Learned alternatives such as Dataset2Vec (Jomaa et al., 2021) and the Neural Statistician (Edwards & Storkey, 2017) learn dedicated dataset encoders from scratch on collections of datasets, without leveraging the representations acquired by large-scale predictive pretraining. This raises a more general question: can the representations learned through large-scale predictive pretraining be repurposed into reusable representations of entire supervised datasets?

We answer this question with TABMF, a foundation model for supervised dataset representation. Our key idea is to repurpose pretrained tabular foundation models (TFMs) such as TabPFN (Hollmann et al., 2025; Grinsztajn et al., 2025) and TabICL (Qu et al., 2025; 2026). These models already process an entire labeled table jointly and learn rich internal representations through predictive pretraining, but are optimized for instance-level prediction rather than for representing the supervised dataset itself. TABMF equips a pretrained TFM with an architecture-aware dataset-level readout and adapts the entire model end-to-end with a contrastive objective over stochastic views of synthetic supervised datasets. The views vary sampled observations, observed variables, and label-preserving feature transformations, encouraging a representation that is stable across different realizations of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

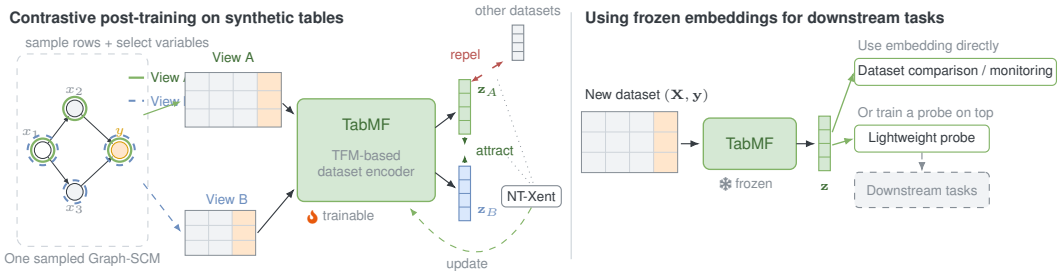


Figure 1: **TABMF overview. Left:** A synthetic supervised dataset is sampled from one Graph-SCM, and two stochastic views vary the sampled observations, observed variables, and label-preserving feature transformations while sharing the same designated target. TABMF encodes both views and is trained contrastively to bring their representations together while separating representations of other datasets. **Right:** At inference time, a frozen TABMF encoder maps a new supervised dataset (X, y) to a fixed-dimensional representation that can be used directly for dataset comparison and monitoring or as input to lightweight downstream probes.

the same supervised dataset. We instantiate this general recipe on both TabICLv2 and TabPFN-2.5 using readouts tailored to their internal architectures (Section 3.2). At inference time, TABMF maps one supervised dataset to one fixed-dimensional vector in a single forward pass.

A surprising finding of our study is that predictive pretraining alone already induces useful dataset-level representations. Simply mean-pooling internal states of frozen pretrained TFMs yields strong dataset embeddings without any dataset-level adaptation. In particular, frozen TFM representations outperform dedicated learned dataset encoders on retrieval and remain competitive with learned encoders and classical meta-features on dataset-property prediction (Section 5.5). This reveals an emergent dataset-level representation capability in predictive TFMs.

We evaluate frozen and adapted representations on real datasets unseen during training. TABMF achieves near-perfect real-data view discrimination and substantially improves retrieval robustness under a broad suite of 19 dataset transformations, with particularly large gains under severe feature subsampling. Beyond retrieval, the learned geometry supports higher-level applications including compact benchmark construction and distribution-shift severity assessment. Across the six primary evaluation tasks, the two TABMF encoders obtain the two strongest aggregate ranks among the evaluated learned and classical dataset representations (Section 5).

Our contributions are:

- **TABMF.** We introduce an end-to-end framework that repurposes pretrained TFMs as zero-shot encoders of supervised datasets through architecture-aware readouts and contrastive post-training on synthetic datasets. At inference time, TABMF maps a previously unseen supervised dataset directly to a fixed-dimensional representation in a single forward pass (Section 3).

- **Emergent dataset representations in predictive TFMs.** We show that predictive pretraining alone already induces strong dataset-level representations. Simple frozen readouts outperform dedicated learned dataset encoders on retrieval and remain competitive on dataset-property prediction (Section 5.5).

- **An extensive evaluation of dataset representations.** We compare frozen and adapted TFM representations with learned dataset encoders and classical meta-features across dataset-property prediction, robust retrieval, compact benchmark construction, and distribution-shift assessment. The two TABMF encoders achieve the strongest aggregate performance across the six-task primary suite (Section 5).

2 Related Work

Meta-features and learned dataset encoders. Hand-crafted meta-features summarize a dataset with fixed statistics (Alcobaça et al., 2020) and underpin algorithm selection (McElfresh et al., 2023), outlier-model selection (Zhao et al., 2021), and zero-shot AutoML (Öztürk et al., 2022; Pineda-Arango et al., 2024). As fixed descriptors, they cannot adapt their representation to the task consuming it, and Herre et al. (2026) report limited ability to explain performance differences between modern tabular foundation models. Some meta-feature families also use explicit target statistics or fit predictive models to the dataset, which can overlap with downstream properties defined through predictive behavior; we control this explicitly for epiplexity prediction (Section B.4.4). Learned alternatives train dataset encoders from scratch: Dataset2Vec (Jomaa et al., 2021) distinguishes batches of the same versus different real datasets with a hierarchical set network, and the Neural Statistician (Edwards & Storkey, 2017) learns a variational summary of a set of exchangeable observations. TabMF differs in starting from a pretrained TFM rather than a randomly initialized set network, and in using synthetic supervised datasets for dataset-level post-training; both baselines are included in our primary comparison (Section 5). Task2Vec (Achille et al., 2019) embeds tasks via Fisher information of a probe network, which requires fitting the probe on every new dataset; our encoders require only a forward pass. Kunz et al. (2026) align numeric tables through their statistics, without learning and without supervised-task structure.

Tabular foundation models and their representations. TabPFN (Hollmann et al., 2025; Grinsztajn et al., 2025) and TabICL (Qu et al., 2025; 2026) are Transformers pretrained on synthetic data for in-context label prediction; TabDPT (Ma et al., 2025) trains on real data, and CARTE (Kim et al., 2024) incorporates column-name semantics. Although these models are designed for prediction, their internal representations have also been reused: Ye et al. (2025) study frozen TabPFN v2 row-level features, while Regenwetter et al. (2026) pool TabPFN-2.5 activations into dataset descriptors for curating synthetic pretraining tasks. We repurpose TabICLv2 and TabPFN-2.5 as encoders of supervised datasets using architecture-aware readouts and contrastive dataset-level post-training (Section 3.2), and separately evaluate what their frozen readouts already capture (Section 5.5).

Contrastive learning and table embeddings. Contrastive methods on tabular data corrupt features (Bahri et al., 2022) or subset them (Ucar et al., 2021) to construct positive views of individual rows, following the instance-level augmentation paradigm of SimCLR (Chen et al., 2020). TabMF instead applies contrastive learning at the dataset level: positive pairs are stochastic views of the same supervised dataset that can differ in their sampled observations, observed variables, and label-preserving feature transformations (Section 3.1).

A separate line of work learns table representations for content-based retrieval and data discovery, including HyTrel (Chen et al., 2023), the TEmBed benchmark (Vogel et al., 2026; Poostforoushan et al., 2026), and generalist tabular embeddings (Qiang et al., 2026). This setting is distinct from ours: TabMF represents a supervised dataset (X, y), with the designated target forming part of the represented object, rather than a target-agnostic table. We nevertheless compare against HyTrel and report TEmBed results separately in Appendix B.7 to examine how supervised dataset representations transfer to content-oriented table tasks.

3 TabMF

Let $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ denote a supervised classification dataset with T observations, H features, and C classes, where $\mathbf{X} \in \mathbb{R}^{T \times H}$ and $\mathbf{y} \in \{0, \ldots, C-1\}^T$. We seek an encoder f_θ that maps a previously unseen supervised dataset to a fixed-dimensional representation $\mathbf{z} = f_\theta(\mathcal{D}) \in \mathbb{R}^{d_z}$. The designated target $\mathbf{y}$ is part of the represented object, so the embedding is intended to capture predictive structure between features and target rather than target-agnostic table content.

TabMF repurposes a pretrained predictive tabular foundation model (TFM) as such a dataset encoder. The general recipe combines three components: (i) a pretrained TFM backbone that contextualizes an entire labeled dataset, (ii) an architecture-aware readout that reduces its internal states to a single dataset representation, and (iii) contrastive post-training over stochastic views of synthetic supervised datasets. The complete encoder is fully differentiable, allowing the backbone, readout, and projection network to be optimized jointly. At inference time, a dataset is mapped directly to a fixed-dimensional embedding in a single forward pass.

We instantiate this recipe on TabICLv2 (Qu et al., 2026) and TabPFN-2.5 (Grinsztajn et al., 2025). We refer to the resulting models as TABMF-ICL and TABMF-PFN, respectively, for brevity. The two models organize their internal representations differently: TabICLv2 reduces observations to fixed-dimensional row states before applying cross-row in-context attention, whereas TabPFN-2.5 maintains contextual representations jointly over observations and feature groups. We therefore use a different dataset-level readout for each backbone (Section 3.2).

We also evaluate dataset-level readouts of the frozen backbones to measure what predictive pretraining provides before contrastive post-training; implementation details are given in Appendix A.1.

3.1 CONTRASTIVE DATASET-LEVEL ADAPTATION

TABMF shapes dataset representations by encouraging consistency across different views of the same supervised dataset. We draw B synthetic supervised datasets $\mathcal{D}_i = (\mathbf{X}_i, \mathbf{y}_i) \sim p_{\text{syn}}$, $i = 1, \ldots, B$, from the graph-structured prior of TabICLv2 (Qu et al., 2026).

For each dataset, we independently construct two variable-sized views. Rows are resampled with replacement, and features are subsampled without replacement. Subsets of the selected features may also undergo noise or Fourier transformations, while their labels remain unchanged. Let $\mathcal{I}_i^v$ and $\mathcal{J}_i^v$ denote the sampled row and feature indices, and let a_i^v denote the feature transformation. For $v \in \{A, B\}$, the resulting dataset view is

$$\mathcal{D}_i^v = \left(a_i^v(\mathbf{X}_i[\mathcal{I}_i^v, \mathcal{J}_i^v]), \mathbf{y}_i[\mathcal{I}_i^v]\right). \tag{1}$$

The views can differ in their observations, observed variables, feature values, and sizes while retaining the same designated prediction target. Sampling ranges and transformation parameters are specified in Section 4.

The shared encoder maps the views to $\mathbf{z}_i^A = f_\theta(\mathcal{D}_i^A)$ and $\mathbf{z}_i^B = f_\theta(\mathcal{D}_i^B)$. The two views of a source dataset form a positive pair; views of different source datasets in the batch form negatives. Positive pairs are defined by their shared source dataset, rather than by matching parameters of the synthetic generator, which are not provided to the encoder.

The architecture-aware readout produces a dataset representation $\mathbf{u}$. A projection network consisting of LayerNorm followed by a two-layer GELU MLP maps it to the normalized embedding $\mathbf{z} = g_\phi(\mathbf{u})/\|g_\phi(\mathbf{u})\|_2$.

We train TABMF with the normalized temperature-scaled cross-entropy objective (NT-Xent) (Chen et al., 2020). For an anchor $\mathbf{z}_i^v$, let $\mathbf{z}_i^{\bar{v}}$ denote the embedding of its other view. The loss is

$$\ell_i^v = -\log \frac{\exp(\text{sim}(\mathbf{z}_i^v, \mathbf{z}_i^{\bar{v}})/\tau)}{\displaystyle\sum_{(j,w)\neq(i,v)} \exp\big(\text{sim}(\mathbf{z}_i^v, \mathbf{z}_j^w)/\tau\big)}, \tag{2}$$

where $j \in \{1, \ldots, B\}$ indexes source datasets and $w \in \{A, B\}$ indexes views. The denominator includes all $2B$ views except the anchor; $\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b}$ for normalized embeddings, and τ is the temperature. We average the loss over both views of all B datasets.

3.2 ARCHITECTURE-AWARE TABMF ENCODERS

Both encoders use the contrastive objective described above, but their dataset-level readouts follow the structure of their respective backbones. Figure 5 in Appendix C illustrates the two designs.

For the TABMF variant with the TabICLv2 backbone, we combine each row state with its target embedding, $\widetilde{\mathbf{r}}_t = \mathbf{r}_t + \text{YEnc}(y_t)$, and append a learned summary token with no associated input observation or target label $\mathbf{R} = [\widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_T, \mathbf{s}_{\text{sum}}]$. We take its final state as the dataset representation:

$$\mathbf{u}_{\text{TabICL}} = \text{LN}_{\text{ICL}}\left(\text{TF}_{\text{ICL}}^{(T)}(\mathbf{R})\right)_{T+1}. \tag{3}$$

For the variant with the TabPFN-2.5 backbone, we denote the feature-group states as $\mathbf{F} \in \mathbb{R}^{T\times M\times d_h}$, where M is the number of feature-group positions and d_h is their hidden dimension. We average each group over observations, $\bar{\mathbf{f}}_m = T^{-1}\sum_{t=1}^{T} \mathbf{F}_{t,m}$, then apply a learned Transformer

across the resulting group states, $\widetilde{\mathbf{F}} = \mathrm{TF}_{\mathrm{col}}(\bar{\mathbf{F}})$, where $\bar{\mathbf{F}}$ collects $\bar{\mathbf{f}}_1, \ldots, \bar{\mathbf{f}}_M$. Writing $\mathcal{M}_{\mathrm{active}}$ for the set of non-padded groups, we obtain the dataset representation as

$$\mathbf{u}_{\mathrm{TabPFN}} = \frac{1}{|\mathcal{M}_{\mathrm{active}}|} \sum_{m \in \mathcal{M}_{\mathrm{active}}} \widetilde{\mathbf{f}}_m. \tag{4}$$

4 Experimental Setup

4.1 Training and Models

Training. We generate synthetic datasets on the fly using the TabICLv2 Graph-SCM prior (Qu et al., 2026), with 5,000–50,000 rows, 5–100 features, and at most ten classes, retaining the remaining prior settings. Each dataset yields two stochastic views as described in Section 3.1, containing 500–5,000 rows and 5–50 features for TabMF-ICL, or 200–1,500 rows and 5–30 features for TabMF-PFN, capped at the source dimensions. The smaller TabMF-PFN views reflect its higher memory requirements. Training views additionally apply noise and Fourier transformations to 30% of the selected columns, with noise standard deviation 0.1 and 16 Fourier bins.

We jointly optimize the backbone, architecture-aware readout, and projection network using AdamW with a global batch size of 64 datasets, a learning rate of 10^{-5}, zero weight decay, and a cosine learning-rate schedule with 5% warmup. The evaluated TabMF-ICL and TabMF-PFN checkpoints use 45,000 optimization steps. Training uses two NVIDIA H100 GPUs per encoder and requires approximately 315 GPU-hours for TabMF-ICL and 273 GPU-hours for TabMF-PFN.

Encoders and baselines. We evaluate the summary-token and column-aware encoders defined in Section 3.2, instantiated with TabICLv2 (Qu et al., 2026) and TabPFN-2.5 (Grinsztajn et al., 2025). TabICL ICL-mean and TabPFN mean-pooling provide frozen-backbone baselines; comparisons with the adapted encoders therefore include changes in readout. We also evaluate alternative TabICL readouts and random initialization, alongside PyMFE (Alcobaça et al., 2020), Dataset2Vec (Jomaa et al., 2021), the Neural Statistician (Edwards & Storkey, 2017), and HyTrel (Chen et al., 2023).

4.2 Evaluation Tasks

We evaluate transfer from synthetic post-training datasets to previously unseen real datasets across the following dataset-level tasks.

Bootstrap view discrimination. On 191 TabBench datasets (Neuralk-AI, 2025), we test whether views of the same dataset have higher cosine similarity than views of different datasets. Four views are generated per dataset by independently resampling rows and subsampling features, with 500–5,000 rows and 5–50 features, capped at the source dimensions. ROC-AUC measures discrimination between same-dataset and cross-dataset view pairs. Per-dataset hard margins additionally measure separation from the closest competing views.

Dataset-property prediction. We probe embeddings for two properties on TabBench. Domain discrimination predicts eight manually assigned domains with a multiclass logistic-regression probe and reports macro one-vs-rest ROC-AUC. Epiplexity measures the structural information that can be extracted from a dataset by a learner of fixed computational capacity (Finzi et al., 2026). For epiplexity prediction, we use prequential estimates obtained with an online logistic classifier as reference values. We train a ridge regressor to predict these estimates from dataset embeddings and evaluate Spearman correlation on held-out datasets. PyMFE uses a leakage-controlled feature set for epiplexity (Appendix B.4.4).

Appendix B.4 specifies preprocessing, coverage, and paired analyses. The epiplexity estimator and additional probes are described in Appendix B.4.3.

Retrieval under transformations. A transformed dataset retrieves its source from a gallery of original datasets by cosine similarity; encoder-specific coverage is detailed in Appendix B.2. The 19 transformations cover row and feature subsampling, recoding, rescaling, missingness, noise, and quantization. Mean recall@5 across transformations measures preservation of dataset identity as the observed table changes.

Table 1: Primary results across six evaluation tasks. Scores are real-data bootstrap AUC, domain macro-OVR AUC, epiplexity regression Spearman, mean retrieval recall@5, TabRepo three-dataset K-means ranking Spearman, and TableShift shift-severity Spearman. Encoders are ordered by their median rank across the six tasks. Higher scores and lower ranks are better; bold marks numerical leaders. Epiplexity uses available-case scores and leakage-controlled PyMFE features; paired common-population results are in Appendix B.4.3.

	TabBench				TabRepo	TableShift	
Encoder	Views	Domain	Epiplexity	Recall@5	Selection	Severity	Rank ↓
TabMF-PFN	**0.9985**	0.683	0.597	**0.965**	**0.929**	0.917	**1.5**
TabMF-ICL	0.9981	**0.702**	0.639	0.938	0.854	**0.933**	2.0
TabICL ICL-mean	0.9775	0.670	**0.742**	0.889	0.575	0.850	3.5
PyMFE	0.9812	0.661	0.106	0.931	0.709	0.733	4.0
TabPFN mean-pooling	0.9414	0.650	0.741	0.821	0.850	0.583	5.0
Dataset2Vec	0.9790	0.643	0.536	0.750	0.652	0.617	5.5
Neural Statistician	0.7917	0.554	0.239	0.677	0.738	−0.150	7.0

Compact benchmark selection. Following Gusev & Zaytsev (2026), we select three datasets and measure how closely their classifier ranking matches that of the full benchmark. We use 144 classification datasets and ten default classifiers from TabRepo (Salinas & Erickson, 2024). Each of 200 shared trials draws a candidate pool containing 80% of the population. K-means selects a dataset representative from each cluster; cosine farthest-first provides a second selector. The metric is mean Spearman correlation with the full-population classifier ranking (Appendix B.3).

Distribution-shift severity. On nine TableShift tasks (Gardner et al., 2023), we test whether ID–OOD embedding distances order shifts consistently with their published mean accuracy drops across estimators. Encoders receive the same label-stratified samples of at most 8,192 rows from each official test split, and preprocessing is fitted on the training split. We compute Spearman correlation between cosine distance and the mean ID-minus-OOD accuracy gap, with one observation per task; the nine tasks, severity targets, and sampled population sizes are reported in Table 16.

5 Results

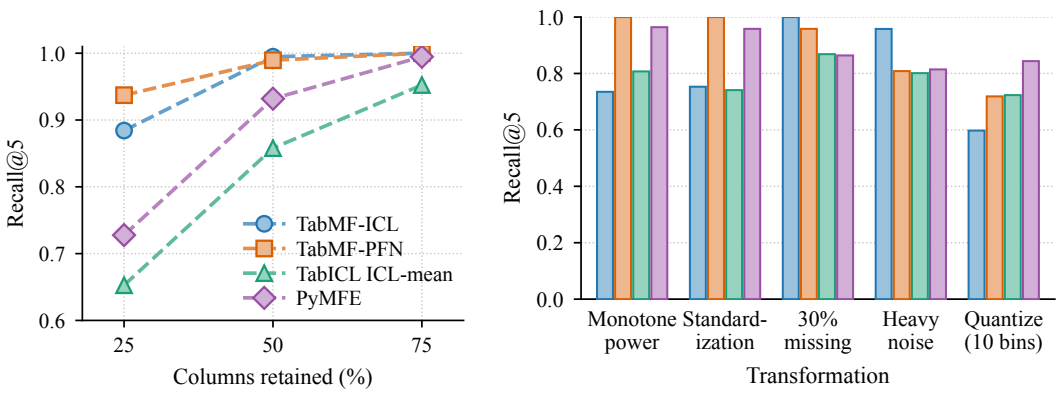
We evaluate whether TabMF provides a reusable representation of previously unseen supervised datasets across several forms of dataset-level reasoning. We first summarize performance across the six primary evaluation tasks, then examine robustness to alternative views of the same dataset and applications of the learned geometry to compact benchmark construction and distribution-shift assessment. Finally, we analyze how much dataset-level structure is already present in frozen predictive TFMs and examine the effect of encoder design choices.

We select one variant per backbone using bootstrap-view discrimination on 50 synthetic datasets generated with the TabICLv2 Graph-SCM prior (Qu et al., 2026). The selected summary-token and column-aware encoders are used throughout; Appendix B.1 gives the candidate comparison. This synthetic benchmark is used only for variant selection and does not enter the primary evaluation or aggregate ranking. Table 1 summarizes the six primary evaluations on real datasets.

5.1 TabMF Provides Strong Dataset Representations

Across the six primary evaluation tasks, the two TabMF variants obtain the two strongest aggregate ranks, with median ranks of 1.5 for TabMF-PFN and 2.0 for TabMF-ICL (Table 1). A TabMF encoder is the numerical leader on five of the six individual tasks, spanning real-data view discrimination, domain prediction, robust retrieval, compact benchmark selection, and distribution-shift severity. The exception is epiplexity prediction, where frozen TFM representations are stronger under the primary linear probe; we return to this result in Section 5.5.

These results indicate that the same learned representation can support multiple dataset-level objectives rather than being useful only for the contrastive task used during training. We additionally examine computational scaling in Appendix B.8, where TabMF produces dataset representations



(a) Recall@5 vs. fraction of columns retained.

(b) Recall@5 under value-level transformations.

Figure 2: Dataset identity remains recoverable with incomplete features, but robustness depends on the encoder and transformation. **(a)** Both TABMF variants outperform ICL-mean and PyMFE when only 25% of columns remain. **(b)** TABMF-PFN better tolerates monotone transformations and standardization, while TABMF-ICL better tolerates heavy noise and missingness; quantization limits both variants. Values are retrieval recall@5; the complete transformation results appear in Figure 3.

substantially faster than classical meta-feature extraction on large tables while adding little inference overhead relative to the corresponding frozen TFM representations. We next examine the strongest effects in more detail.

5.2 TABMF LEARNS A ROBUST GEOMETRY FOR DATASET COMPARISON

Both TABMF encoders transfer the separation learned from synthetic training views to unseen real datasets. On real-data bootstrap-view discrimination, TABMF-ICL and TABMF-PFN achieve ROC-AUCs of 0.9981 and 0.9985, respectively (Table 1). Because several baselines also obtain high AUC, we additionally evaluate the harder per-dataset margin criterion. Both TABMF variants obtain significantly larger hard margins than all nine baselines in this comparison (Table 4); the margin construction and paired tests are detailed in Appendix B.1.

The advantage persists under substantially broader transformations. When recall@5 is averaged across applicable transformations within each dataset, both TABMF variants significantly outperform TabICL ICL-mean, while TABMF-PFN additionally improves significantly over PyMFE (Appendix B.2). The clearest shared advantage occurs under severe feature removal: when only 25% of features remain, recall@5 is 0.884 and 0.937 for TABMF-ICL and TABMF-PFN, compared with 0.653 for TabICL ICL-mean and 0.728 for PyMFE.

The two variants show different robustness profiles (Figure 2). TABMF-PFN better preserves dataset identity under monotone transformations and standardization, whereas TABMF-ICL is more robust to heavy noise and missingness. Quantization remains a shared limitation, with both variants falling below PyMFE. Overall, TABMF substantially improves robustness to alternative views of the same dataset, particularly when much of the original feature set is unavailable.

5.3 COMPACT BENCHMARK SELECTION PRESERVES CLASSIFIER RANKINGS

We next test whether the learned geometry can support a dataset-level decision task that is not used during training. At a three-dataset budget, TABMF-PFN achieves the strongest classifier-ranking correlation on TabRepo, reaching 0.929 with K-means selection compared with 0.850 for frozen TabPFN mean-pooling and 0.777 for random selection (Table 1).

The result is also consistent across the two evaluated selection rules at the three-dataset budget. Under cosine farthest-first selection, TABMF-PFN reaches 0.823, compared with 0.673 for frozen TabPFN mean-pooling and 0.777 for random selection. TABMF-ICL also exceeds random selection under both selectors, reaching 0.854 with K-means and 0.815 with farthest-first. The result is robust

across nearby compact selection budgets: for each $k \in \{2, 3, 4\}$, a TABMF variant obtains the highest numerical score under both K-means and farthest-first selection. The relative ordering of TABMF-ICL and TABMF-PFN varies with the selection budget, and Appendix B.3 reports the full sensitivity analysis.

5.4 EMBEDDING DISTANCES TRACK DISTRIBUTION-SHIFT SEVERITY

The learned representation also provides a meaningful notion of distance between related dataset distributions. Across the nine TableShift tasks, the Spearman correlation between ID–OOD embedding distance and the published mean accuracy drop is 0.933 for TABMF-ICL and 0.917 for TABMF-PFN (Table 1). The strongest baseline, TabICL ICL-mean, obtains 0.850.

This evaluation complements transformation retrieval. Retrieval rewards invariance to changes that should preserve dataset identity, whereas TableShift rewards sensitivity to population changes associated with deteriorating predictive performance. The high observed correlations therefore indicate that the learned geometry can be invariant to nuisance transformations while remaining sensitive to consequential distribution shifts. Because the evaluation contains only nine task-level observations, we do not interpret the numerical ordering as evidence of pairwise superiority between encoders; Appendix B.5 summarizes the statistical scope of the aggregate evaluation.

5.5 PREDICTIVE TFMS ALREADY ENCODE DATASET-LEVEL INFORMATION

Having established the performance of TABMF, we ask how much of this dataset-level capability is already induced by predictive pretraining. The answer is: a substantial part. Without any dataset-level optimization, TabICL ICL-mean achieves a mean retrieval recall@5 of 0.889, outperforming Dataset2Vec (0.750) and the Neural Statistician (0.677), while remaining close to PyMFE (0.931). Frozen TFM representations also encode substantial dataset-property information: TabICL ICL-mean and TabPFN mean-pooling obtain domain AUCs of 0.670 and 0.650 and epiplexity correlations of 0.742 and 0.741, respectively.

We use tuned linear probes for the primary property evaluations because they provide a simple, controlled measure of how readily each property is accessible from a representation. Under these probes, dataset-level post-training does not uniformly improve property recoverability. Although TABMF-ICL obtains the highest observed domain AUC (0.702), the paired analysis does not establish an advantage over the corresponding frozen readouts (Appendix B.4.2). For continuous epiplexity prediction, both frozen encoders obtain higher correlations than their TABMF counterparts, with three of the four cross-comparisons significant after correction (Appendix B.4.3).

These primary comparisons characterize linear accessibility rather than total information content. We therefore repeat both evaluations with substantially more expressive nonlinear probes as a capacity check. With a tuned RBF-and-polynomial SVM, TABMF-ICL obtains a higher domain AUC than frozen TabICL (0.700 versus 0.644), and the corresponding paired per-dataset log-loss comparison is significant after Holm correction ($p = 0.0278$ over the full 11-encoder comparison pool). The corresponding PFN comparison remains unresolved (Appendix B.4.2). With a strengthened multi-kernel ridge probe for epiplexity, neither adapted encoder is significantly separated from its corresponding frozen representation under the same full-pool correction scope. The complete epiplexity results are reported in Appendix B.4.3.

Together, these results show that predictive TFM pretraining itself induces useful representations of entire supervised datasets. Dataset-level post-training strongly improves the geometry for robust dataset comparison, but can also change how individual properties are organized in the representation space. The linear results should therefore be read as differences in readily accessible structure; the nonlinear checks show that the apparent ties and losses do not, by themselves, establish that the corresponding information is absent from the adapted representations.

5.6 ABLATIONS

Table 2 compares the evaluated initialization, objective, and readout variants using mean retrieval recall@5. Pretrained initialization substantially outperforms scratch initialization for both backbones under matched architecture, readout, and NT-Xent training: recall increases from 0.797 to 0.938

Table 2: Encoder design comparisons using mean retrieval recall@5. Each block compares configurations within one backbone across initialization, readout, and objective choices.

Backbone	Readout	Initialization	Objective	Recall@5
TabICL	Summary-token	Pretrained	NT-Xent	**0.938**
TabICL	Summary-token	Scratch	NT-Xent	0.797
TabICL	Summary-token	Pretrained	Triplet	0.924
TabICL	Summary-token	Pretrained	Exp. distance	0.883
TabPFN	Column-attention	Pretrained	NT-Xent	**0.965**
TabPFN	Column-attention	Scratch	NT-Xent	0.587
TabPFN	Summary-token	Pretrained	NT-Xent	0.935
TabPFN	Pseudo-test	Pretrained	NT-Xent	0.957
TabPFN	Column-attention	Pretrained	Triplet	0.962
TabPFN	Column-attention	Pretrained	Exp. distance	0.920

for the TabICL summary-token encoder and from 0.587 to 0.965 for the TabPFN column-aware encoder.

Among the evaluated objectives, NT-Xent obtains the highest recall for both selected architectures: 0.938 versus 0.924 for triplet and 0.883 for exponential distance with TABMF-ICL, and 0.965 versus 0.962 and 0.920, respectively, with TABMF-PFN.

For TabPFN-2.5, the column-aware readout with NT-Xent obtains 0.965 recall, compared with 0.935 for the summary-token readout and 0.957 for the pseudo-test variant.

6 Discussion and Limitations

Our findings show that predictive TFMs can support reusable dataset representations across several forms of dataset-level reasoning. We deliberately focus on adapting existing pretrained TFMs with a limited additional training budget. Pretrained initialization outperforms training from scratch for both backbones, supporting this practical reuse strategy. The representations are explicitly supervised: changing the designated target can change the representation even when the feature table is fixed. Our evaluation covers classification datasets and two pretrained TFM backbones; extending the framework to regression and other architectures would further establish its generality. Post-training uses the TabICLv2 Graph-SCM prior, and the resulting encoders transfer to unseen real datasets. Broader synthetic priors or mixtures of synthetic and real datasets offer a direction for extending this coverage. Different applications may also require different notions of dataset similarity, motivating representations whose geometry can be conditioned on the downstream objective.

7 Conclusion

We introduced TABMF, a framework for repurposing pretrained tabular foundation models as encoders of entire supervised datasets. Through architecture-aware dataset readouts and contrastive post-training on synthetic dataset views, TABMF maps an unseen supervised dataset to a fixed-dimensional representation in a single forward pass. Across six primary evaluations on real datasets, the two TABMF instantiations obtain the strongest aggregate performance among the evaluated learned and classical dataset representations, with particularly strong results in robust dataset comparison, compact benchmark construction, and distribution-shift assessment.

We additionally find that predictive TFM pretraining alone already induces useful dataset-level representations, even without explicit dataset-level training. Together, these results show that tabular foundation models can serve not only as predictors over individual observations, but also as a foundation for learning reusable representations of supervised datasets.

AI USE STATEMENT

The research idea, methodological direction, and experimental design were developed by the authors. Generative AI was used to assist with identifying one of the benchmarks used in the evaluation, with coding parts of the project, and with drafting and polishing portions of the manuscript. All AI-assisted code and manuscript content were reviewed by the authors, and the resulting implementations, experiments, analyses, and claims were checked for correctness. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human-subject experimentation or the collection of new personal data. Our experiments use existing benchmark datasets and synthetic datasets generated from a tabular foundation-model prior. As with other learned representations, downstream applications may reflect biases present in the underlying data; we do not evaluate deployment in high-stakes decision-making settings.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce our method and evaluation throughout the main paper and appendices. Section 3 specifies the TABMF training objective and architecture-aware dataset readouts, while Section 4 describes the training setup, synthetic data generation, evaluated models, and primary evaluation protocols. Appendix A provides additional details on the frozen representations, alternative encoder variants, and metric-learning objectives. Appendix B documents the evaluation populations, preprocessing, statistical procedures, transformation definitions, probe protocols, benchmark-selection sensitivity analyses, and task-level inputs used in the reported experiments. Anonymized source code is available at `https://anonymous.4open.science/r/TabMF-ICLR`.

REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task Embedding for Meta-Learning. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Edesio Alcobaça, Felipe Siqueira, Adriano Rivolli, Luís P. F. Garcia, Jefferson T. Oliva, and André C. P. L. F. de Carvalho. MFE: Towards Reproducible Meta-Feature Extraction. *Journal of Machine Learning Research*, 21(111):1–5, 2020.

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. SCARF: Self-Supervised Contrastive Learning Using Random Feature Corruption. In *International Conference on Learning Representations*, 2022.

Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. HyTrel: Hypergraph-enhanced Tabular Data Representation Learning. In *Advances in Neural Information Processing Systems*, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, 2020.

Harrison Edwards and Amos Storkey. Towards a Neural Statistician. In *International Conference on Learning Representations*, 2017.

Marc Finzi, Shikai Qiu, Yiding Jiang, Pavel Izmailov, J. Zico Kolter, and Andrew Gordon Wilson. From entropy to epiplexity: Rethinking information for computationally bounded intelligence, 2026. URL `https://arxiv.org/abs/2601.03220`.

Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with tableshift. *Advances in Neural Information Processing Systems*, 2023.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models. 2025. arXiv:2511.08667.

Rostislav Gusev and Alexey Zaytsev. Benchmarking on tasks that matter: Dataset selection for preserving model rankings. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 8954–8965. ACM, August 2026. doi: 10.1145/3770855.3817569. URL http://dx.doi.org/10.1145/3770855.3817569.

Markus Herre, Andrej Tschalzev, Sascha Marton, and Christian Bartelt. Explaining Tabular Foundation Model Differences Through Meta-Features. 2026. arXiv:2605.28418.

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

Madelon Hulsebos, Çağatay Demiralp, and Paul Groth. GitTables: A Large-Scale Corpus of Relational Tables. *Proceedings of the ACM on Management of Data*, 1(1), 2023.

Hadi S. Jomaa, Lars Schmidt-Thieme, and Josif Grabocka. Dataset2Vec: Learning Dataset Meta-Features. *Data Mining and Knowledge Discovery*, 35:964–985, 2021.

Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. CARTE: Pretraining and Transfer for Tabular Learning. In *International Conference on Machine Learning*, 2024.

M. Ross Kunz, John Merickel, and Keith Wilson. Statistical Embeddings for Similarity, Retrieval, and Interpretable Alignment of Numeric Tabular Datasets. 2026. arXiv:2605.30289.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L. Caterini. TabDPT: Scaling Tabular Foundation Models on Real Data. In *Advances in Neural Information Processing Systems*, 2025.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Benjamin Feuer, Chinmay Hegde, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When Do Neural Nets Outperform Boosted Trees on Tabular Data? In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

Neuralk-AI. Tabbench: A tabular machine learning benchmark. 2025.

Ekrem Öztürk, Fabio Ferreira, Hadi S. Jomaa, Lars Schmidt-Thieme, Josif Grabocka, and Frank Hutter. Zero-Shot AutoML with Pretrained Models. In *International Conference on Machine Learning*, 2022.

Sebastian Pineda-Arango, Fabio Ferreira, Arlind Kadra, Frank Hutter, and Josif Grabocka. QuickTune: Quickly Learning Which Pretrained Model to Finetune and How. In *International Conference on Learning Representations*, 2024.

Ayeen Poostforoushan, Liane Vogel, and Carsten Binnig. TEmBed-T: A Multi-Dimensional Benchmark for Table-Level Embeddings. In *4th International Workshop on Tabular Data Analysis (TaDA) @ VLDB*, 2026. arXiv:2607.24130.

Minjie Qiang, Mingming Zhang, Xiaoyi Bao, Xing Fu, Yu Cheng, Weiqiang Wang, Zhongqing Wang, and Ningtao Wang. TabEmbed: Benchmarking and Learning Generalist Embeddings for Tabular Understanding. 2026. arXiv:2605.04962.

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.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A Better, Faster, Scalable, and Open Tabular Foundation Model. 2026. arXiv:2602.11139.

Lyle Regenwetter, Rosen Yu, Cyril Picard, and Faez Ahmed. Engineering Regression Without Real-Data Training: Domain Adaptation for Tabular Foundation Models Using Multi-Dataset Embeddings. 2026. arXiv:2603.04692.

David Salinas and Nick Erickson. Tabrepo: A large scale repository of tabular model evaluations and its automl applications, 2024. URL `https://arxiv.org/abs/2311.02971`.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. SubTab: Subsetting Features of Tabular Data for Self-Supervised Representation Learning. In *Advances in Neural Information Processing Systems*, 2021.

Liane Vogel, Kavitha Srinivas, Niharika D'Souza, Sola Shirai, Oktie Hassanzadeh, and Horst Samulowitz. Towards Universal Tabular Embeddings: A Benchmark Across Data Tasks. 2026. arXiv:2604.21696.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A Closer Look at TabPFN v2: Understanding Its Strengths and Extending Its Capabilities. In *Advances in Neural Information Processing Systems*, 2025.

Yue Zhao, Ryan A. Rossi, and Leman Akoglu. Automatic Unsupervised Outlier Model Selection. In *Advances in Neural Information Processing Systems*, 2021.

A Additional Dataset Representations and TabMF Variants

The main text focuses on the pretrained TabICLv2 ICL representation and the two TabMF encoders used in our primary evaluation. Here, we describe the remaining pretrained representations and learned readouts considered in our study, together with the alternative metric-learning objectives.

A.1 Representations from Pretrained TabICLv2

TabICLv2 exposes representations at several stages of its feature-to-row-to-ICL hierarchy (Qu et al., 2026): the column, row, and ICL stages without dataset-level training.

Column mean. Let

$$\mathbf{E} \in \mathbb{R}^{T\times(G+C_{\text{cls}})\times D} \tag{5}$$

denote the output of the pretrained TabICLv2 column embedder, where the token dimension contains the feature-group and reserved CLS positions. The column embeddings are target-aware, i.e., the labels $\mathbf{y}$ are provided to the column embedder. We obtain a dataset representation by averaging jointly over observations and token positions,

$$\mathbf{u}_{\text{col-mean}} = \frac{1}{T(G+C_{\text{cls}})}\sum_{t=1}^{T}\sum_{m=1}^{G+C_{\text{cls}}}\mathbf{E}_{t,m}, \tag{6}$$

followed by ℓ_2 normalization. This representation is extracted before the row-wise and ICL stages of TabICLv2.

Row mean. The row interactor processes the column embeddings and produces a fixed-dimensional representation

$$\mathbf{r}_t \in \mathbb{R}^{C_{\text{cls}}D} \tag{7}$$

for each observation. We aggregate these native row representations as

$$\mathbf{u}_{\text{row-mean}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{r}_t. \tag{8}$$

The column embedder and row interactor use the pretrained TabICLv2 weights; no dataset-level adaptation is performed.

ICL mean. We next extract the representations produced by TabICLv2's native ICL transformer. Starting from the row representations, we incorporate the native target embedding,

$$\widetilde{\mathbf{r}}_t = \mathbf{r}_t + \mathrm{YEnc}(y_t), \tag{9}$$

for datasets whose class cardinality is supported by the pretrained target encoder. The resulting row sequence is processed by the pretrained ICL transformer with all T observations acting as labeled context,

$$\mathbf{Q} = \mathrm{TF}_{\mathrm{ICL}}\left([\widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_T]; \mathrm{train_size} = T\right). \tag{10}$$

We apply the native pre-decoder normalization of the ICL predictor and average the resulting hidden states,

$$\mathbf{u}_{\text{ICL-mean}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{Q}_t, \tag{11}$$

followed by ℓ_2 normalization. The prediction decoder is not used. This is the pretrained representation reported as the primary raw-TFM baseline in the main text.

Randomly initialized control. We additionally evaluate whether the representation quality arises from foundation-model pretraining rather than the architecture and pooling operation alone. This control uses the same column-embedder and row-interactor path as the row-mean encoder, but initializes both modules randomly. The resulting row states are averaged,

$$\bar{\mathbf{r}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t, \tag{12}$$

and passed through a randomly initialized projection network,

$$\mathbf{z}_{\mathrm{random}} = \mathrm{norm}\left(g_\phi(\bar{\mathbf{r}})\right). \tag{13}$$

The ICL transformer is not used in this control. Thus, this baseline corresponds to a randomly initialized column-to-row encoder with the same row-mean readout, rather than a randomly initialized full TabICLv2 ICL stack.

A.2 Alternative TabICLv2 Encoders

In addition to the summary-token encoder described in Sec. 3.2, we study two alternative mechanisms for constructing learned dataset representations from TabICLv2 (Qu et al., 2026).

Contrastively trained row mean. The first learned variant retains the native row-mean aggregation,

$$\mathbf{u}_{\mathrm{mean}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t, \tag{14}$$

but optimizes the TabICLv2 backbone and projection network with the dataset-level contrastive objective. This provides a direct comparison between using pretrained row representations as-is and adapting the same representation to the dataset-level objective while holding the readout fixed.

Column-aware encoder. The native TabICLv2 row representation reduces the feature dimension before dataset-level aggregation. We instead construct the dataset representation from the contextual feature states produced immediately before the final CLS aggregation of the row interactor. Let

$$\mathbf{F}_{\mathrm{TabICL}} \in \mathbb{R}^{T \times M \times D} \tag{15}$$

denote these feature representations after removing the CLS positions.

We first aggregate each feature representation across observations,

$$\bar{\mathbf{f}}_m = \frac{1}{T}\sum_{t=1}^{T} \mathbf{F}_{\mathrm{TabICL},t,m}. \tag{16}$$

We then apply learned self-attention across the resulting dataset-level feature representations,

$$\widetilde{\mathbf{F}} = \mathrm{TF}_{\mathrm{col}}\left([\bar{\mathbf{f}}_1, \ldots, \bar{\mathbf{f}}_M]\right), \tag{17}$$

and pool over the active feature positions,

$$\mathbf{u}_{\mathrm{col}} = \frac{1}{|\mathcal{M}_{\mathrm{active}}|} \sum_{m \in \mathcal{M}_{\mathrm{active}}} \widetilde{\mathbf{f}}_m. \tag{18}$$

This readout therefore preserves the explicit feature dimension through the dataset-level attention stage rather than first collapsing each observation to its CLS representation.

A.3 Alternative TabPFN-2.5 Encoders

TabPFN-2.5 maintains contextual states over observations and feature groups throughout its transformer backbone (Grinsztajn et al., 2025). In addition to the column-aware readout described in Sec. 3.2, we study hierarchical summary-token and pseudo-test representations.

Hierarchical summary-token encoder. TabPFN-2.5 does not expose the same fixed-dimensional row representation followed by a separate ICL transformer as TabICLv2. We therefore learn the feature-to-row and row-to-dataset reductions explicitly.

Let

$$\mathbf{F}_{\mathrm{TabPFN}} \in \mathbb{R}^{T \times M_f \times D} \tag{19}$$

denote the contextual feature-group states of the labeled observations. For each observation t, $K = 4$ shared learned queries attend to its feature representations. Their outputs are concatenated and normalized,

$$\mathbf{c} * t = \mathrm{LN}\left(\mathrm{vec}\left(\mathrm{CrossAttn}\left(\mathbf{Q} * 1 : K, \mathbf{F}_{\mathrm{TabPFN},t,:}\right)\right)\right) \in \mathbb{R}^{KD}. \tag{20}$$

In the evaluated model, $D = 192$ and hence $KD = 768$. The target encoder maps the one-hot class label directly into this same space using a learned linear projection,

$$\mathrm{YEnc}(y_t) = \mathbf{W} * y\,\mathrm{onehot}(y_t) + \mathbf{b} * y \in \mathbb{R}^{KD}. \tag{21}$$

We add this target representation to each row representation and append a learned summary token,

$$\mathbf{C} = [\mathbf{c} * 1 + \mathrm{YEnc}(y_1), \ldots, \mathbf{c} * T + \mathrm{YEnc}(y_T), \mathbf{s} * \mathrm{sum}]. \tag{22}$$

A dedicated transformer operates across the resulting row sequence, and the summary position defines the dataset representation,

$$\mathbf{u}_{\mathrm{sum}} = \mathrm{TF}_{\mathrm{sum}}(\mathbf{C})_{T+1}. \tag{23}$$

Pseudo-test encoder. The pseudo-test representation instead uses TabPFN-2.5's native train–test computation. We sample K observations without replacement and append copies of their feature vectors to the dataset as unlabeled test rows,

$$\mathbf{X}' = \mathrm{concat}\left(\mathbf{X}, \mathbf{X}_{\mathcal{J}}\right), \qquad |\mathcal{J}| = K. \tag{24}$$

The original T observations remain the labeled context, while the appended rows are processed as test observations by the native TabPFN ICL computation.

Let

$$\mathbf{g}_1, \ldots, \mathbf{g}_K \in \mathbb{R}^D \tag{25}$$

denote the target-column hidden states at the resulting pseudo-test positions. We concatenate them to form

$$\mathbf{u}_{\mathrm{ptest}} = [\mathbf{g}_1; \ldots; \mathbf{g}_K] \in \mathbb{R}^{KD}. \tag{26}$$

The concatenated representation is subsequently mapped through the projection network. This construction does not introduce an additional row- or dataset-level attention module; the pseudo-test states are generated directly by TabPFN-2.5's pretrained in-context computation.

A.4 Alternative Metric-Learning Objectives

The main TabMF models use NT-Xent (Chen et al., 2020).

We also study exponential distance and batch-hard triplet objectives while retaining the same dataset views and encoder architectures.

Let

$$\mathbf{z}_i^A, \mathbf{z}_i^B \in \mathbb{R}^{d_z} \tag{27}$$

denote the ℓ_2-normalized representations of the two views of dataset i.

Exponential distance. We define the cross-view Euclidean distance

$$\delta_{ij} = \left\| \mathbf{z}_i^A - \mathbf{z}_j^B \right\|_2 . \tag{28}$$

Pairs with $i = j$ correspond to views of the same dataset, while $i \neq j$ denotes different datasets. The objective is

$$\mathcal{L}_{\text{exp}} = \left(1 - \frac{1}{B} \sum_{i=1}^{B} e^{-\gamma \delta_{ii}} \right) + \frac{1}{B(B-1)} \sum_{i \neq j} e^{-\gamma \delta_{ij}}, \tag{29}$$

where γ controls the decay with distance. The first term attracts the two views of each dataset, while the second penalizes small distances between representations of different datasets.

Batch-hard triplet loss. We additionally evaluate a triplet-margin objective (Schroff et al., 2015). For each anchor

$$\mathbf{a}_i = \mathbf{z}_i^A, \qquad \mathbf{p}_i = \mathbf{z}_i^B, \tag{30}$$

we select the closest representation in the opposite view originating from another dataset,

$$j^\star = \arg\min_{j \neq i} \left\| \mathbf{z}_i^A - \mathbf{z}_j^B \right\|_2 , \qquad \mathbf{n}_i = \mathbf{z}_{j^\star}^B . \tag{31}$$

The batch-hard triplet objective is

$$\mathcal{L}_{\text{triplet}} = \frac{1}{B} \sum_{i=1}^{B} \max\left(0,\ \|\mathbf{a}_i - \mathbf{p}_i\|_2 - \|\mathbf{a}_i - \mathbf{n}_i\|_2 + m\right), \tag{32}$$

where m denotes the triplet margin.

B Additional Results and Statistical Comparisons

B.1 Bootstrap View Discrimination and Variant Selection

We select one variant per backbone using bootstrap-view discrimination AUC on 50 synthetic datasets generated from the TabICLv2 Graph-SCM prior (Qu et al., 2026). Table 3 compares synthetic bootstrap-view discrimination across readouts, training objectives, and initialization schemes. The selected summary-token and column-aware encoders achieve synthetic AUCs of 0.9999 and 0.9996, respectively. Several candidates approach saturation; we therefore use this benchmark for variant selection. The selected summary-token and column-aware encoders achieve synthetic AUCs of 0.9999 and 0.9996, respectively.

Populations and views. The real-data population consists of 191 TabBench datasets. Synthetic datasets are generated with the TabICLv2 prior using 5–100 features, at most ten classes, 5000–50000 rows, and the prior's unpredictable-dataset filters. All encoders are evaluated on the same source datasets and the same four sampled views of each dataset. For each source dataset, we generate four views by sampling rows with replacement and features without replacement. View sizes range from 500–5000 rows and 5–50 features, capped at the source dimensions. We ℓ_2-normalize the resulting embeddings and compute cosine similarities between all distinct view pairs. Pairs of views from the same source dataset are positive; pairs from different source datasets are negative.

Table 3: Synthetic bootstrap-view discrimination AUC across encoder configurations. Row-mean denotes a trained encoder, distinct from the frozen row-mean baseline.

Backbone	Readout	Loss	Initialization	AUC
TabICL	Summary-token	NT-Xent	Pretrained	0.9999
TabPFN	Column-aware	NT-Xent	Pretrained	0.9996
TabICL	Row-mean	NT-Xent	Pretrained	0.9996
TabPFN	Pseudo-test	NT-Xent	Pretrained	0.9995
TabICL	Row-mean	Triplet	Pretrained	0.9994
TabPFN	Pseudo-test	Triplet	Pretrained	0.9994
TabICL	Summary-token	Triplet	Pretrained	0.9992
TabICL	Row-mean	NT-Xent	Scratch	0.9985
TabPFN	Column-aware	Triplet	Pretrained	0.9981
TabICL	Summary-token	NT-Xent	Scratch	0.9974
TabICL	Summary-token	Triplet	Scratch	0.9973
TabICL	Summary-token	Exp. distance	Pretrained	0.9972
TabICL	Row-mean	Triplet	Scratch	0.9960
TabICL	Row-mean	Exp. distance	Pretrained	0.9928

Table 4: Paired comparisons of per-dataset hard margins in bootstrap-view discrimination. Both adapted encoders improve over each of nine baselines in each population. Holm correction is applied across all 18 comparisons within each population. The final column gives an upper bound on the largest adjusted p-value. These tests assess margins, not AUC. Synthetic results characterize the population used for model selection.

Population	Datasets	Significant improvements	Largest Holm p
TabBench	191	18/18	$< 1.6 \times 10^{-16}$
Synthetic	50	18/18	$< 5.0 \times 10^{-14}$

Hard margins. For each source dataset, we define the hard margin as the mean within-dataset pairwise cosine similarity minus the maximum cross-dataset cosine similarity involving that dataset. A positive margin indicates that the average within-dataset similarity exceeds the similarity of the closest competing cross-dataset pair.

We compare both adapted encoders against nine baselines: PyMFE, HyTrel, Dataset2Vec, the Neural Statistician, frozen TabPFN mean-pooling, and four TabICL controls: ICL-mean, row-mean, column-mean, and random initialization. On TabBench, median margins are 0.0809 for TABMF-ICL and 0.0587 for TABMF-PFN, compared with at most 0.0045 among the baselines. On synthetic datasets, the corresponding medians are 0.5945 and 0.5605, compared with at most 0.0306 among the baselines.

Both adapted encoders significantly outperform every baseline in each population under two-sided Wilcoxon signed-rank tests paired by dataset, with Holm correction across all 18 comparisons within each population. Table 4 summarizes these tests. These comparisons assess margins rather than AUC; the synthetic results characterize the population used for model selection.

B.2 RETRIEVAL ROBUSTNESS UNDER DATASET TRANSFORMATIONS

For paired inference, we average recall@5 across applicable transformations within each dataset. Two-sided Wilcoxon signed-rank tests compare these dataset-level scores on the population shared by each encoder pair, dropping zero differences. The four comparisons of the two adapted encoders against TabICL ICL-mean and PyMFE form a single Holm-correction family (Table 6). Both adapted encoders significantly outperform TabICL ICL-mean, while TABMF-PFN also significantly outperforms PyMFE. These tests concern dataset-level averages, whereas the headline retrieval score weights the 19 transformations equally.

Figure 3 reports recall@5 for all 19 transformations.

Table 5: Dataset transformations used for retrieval evaluation. Here, n and d denote the original numbers of observations and features, and μ_j and σ_j denote the mean and standard deviation of numerical feature j. We write $k_r = \max\{\min(n, 20), \mathrm{round}(rn)\}$ and $h_r = \max\{1, \mathrm{round}(rd)\}$. Grouped entries denote separate transformations for each listed parameter value.

Transformation	Definition
Keep 20% / 50% of rows	Sample k_r observations without replacement, $r \in \{0.2, 0.5\}$. Skip when $k_r \geq n$.
Bootstrap rows	Sample n observations with replacement.
Duplicate rows	Retain all observations and append $\max\{1, \mathrm{round}(0.3n)\}$ observations sampled with replacement.
Keep 25% / 50% / 75% of features	Sample h_r features without replacement, $r \in \{0.25, 0.5, 0.75\}$. Skip when $h_r \geq d$.
Duplicate features	Append copies of $h_{0.3}$ features selected without replacement.
Add noisy features	Append $h_{0.3}$ numerical features with independent $\mathcal{N}(0, 1)$ entries.
Light / heavy noise	Add independent $\mathcal{N}(0, (\alpha\sigma_j)^2)$ noise to numerical features, with $\alpha \in \{0.1, 0.5\}$. Features with zero or non-finite standard deviation are skipped.
Affine rescaling	For each numerical feature, apply $x' = a_j x + b_j$, where $\log a_j \sim \mathcal{U}(\log 0.2, \log 5)$ and $b_j \sim \mathcal{N}(0, \sigma_j^2)$.
Monotone power	For each numerical feature, set $t = x/\sigma_j$ and apply $x' = \mathrm{sign}(t)\lvert t\rvert^{a_j}$, where $a_j \sim \mathcal{U}(0.5, 2)$.
Quantization	For numerical features with more than ten distinct observed values, assign integer codes using ten quantile bins of observation ranks. Ties are broken by observation order; existing missing values are preserved.
Standardization	Apply $x' = (x - \mu_j)/\sigma_j$ to numerical features.
10% / 30% missingness	Independently replace each feature entry with a missing value with probability 0.1 or 0.3, including categorical entries.
Categorical recoding	Bijectively relabel the observed categories using a random permutation, for each non-numerical feature with at least two distinct observed values.
Combined view	Sample $k_{0.7}$ observations with replacement and $h_{0.7}$ features without replacement, then apply light numerical noise ($\alpha = 0.1$). Retain the sampled view if numerical noise is inapplicable.

Table 6: Dataset-level paired comparisons of transformation retrieval. For each dataset, recall@5 is averaged across applicable transformations. Entries are Holm-adjusted p-values from two-sided Wilcoxon signed-rank tests, correcting across all four comparisons. Tests pair 190 datasets, except TabMF-PFN versus PyMFE, which pairs 191. Bold denotes a significant improvement of the adapted encoder. These tests assess dataset-level averages; the headline retrieval score instead averages recall@5 equally across transformations.

Baseline	TabMF-ICL	TabMF-PFN
TabICL ICL-mean	$\mathbf{1.0 \times 10^{-7}}$	$\mathbf{5.8 \times 10^{-15}}$
PyMFE	0.522	$\mathbf{1.6 \times 10^{-5}}$

B.3 Three-Dataset Benchmark Selection

We apply the benchmark-selection framework of Gusev & Zaytsev (2026) to TabRepo (Salinas & Erickson, 2024), using 144 classification datasets and ten default classifiers. Each comparison uses 200 shared candidate pools containing 80% of the population and the same full-population classifier ranking as reference. Embeddings are standardized and ℓ_2-normalized before either K-means representative selection or cosine farthest-first selection. Each rule selects three datasets. Table 7 reports the results for both selection rules. Only K-means selection contributes to the primary aggregate rank; farthest-first selection provides an additional evaluation.

Budget sensitivity and statistical tests. We use $k = 3$ as the primary compact-benchmark setting and report sensitivity over $k \in \{2, 3, 4\}$ below. Because $k = 3$ was selected during the evaluation analysis, we interpret the corresponding significance tests together with this budget-sensitivity analysis. Paired Wilcoxon signed-rank tests use the 200 shared candidate-pool trials. For each selection rule, the comparisons of the two adapted encoders against Random, TabICL ICL-mean, PyMFE, Dataset2Vec, the Neural Statistician, and frozen TabPFN mean-pooling form a single 12-comparison Holm-correction family. Table 8 reports the corresponding Holm-adjusted p-values for K-means. The reported 95% confidence intervals are percentile-bootstrap intervals for mean paired differences across candidate-pool trials.

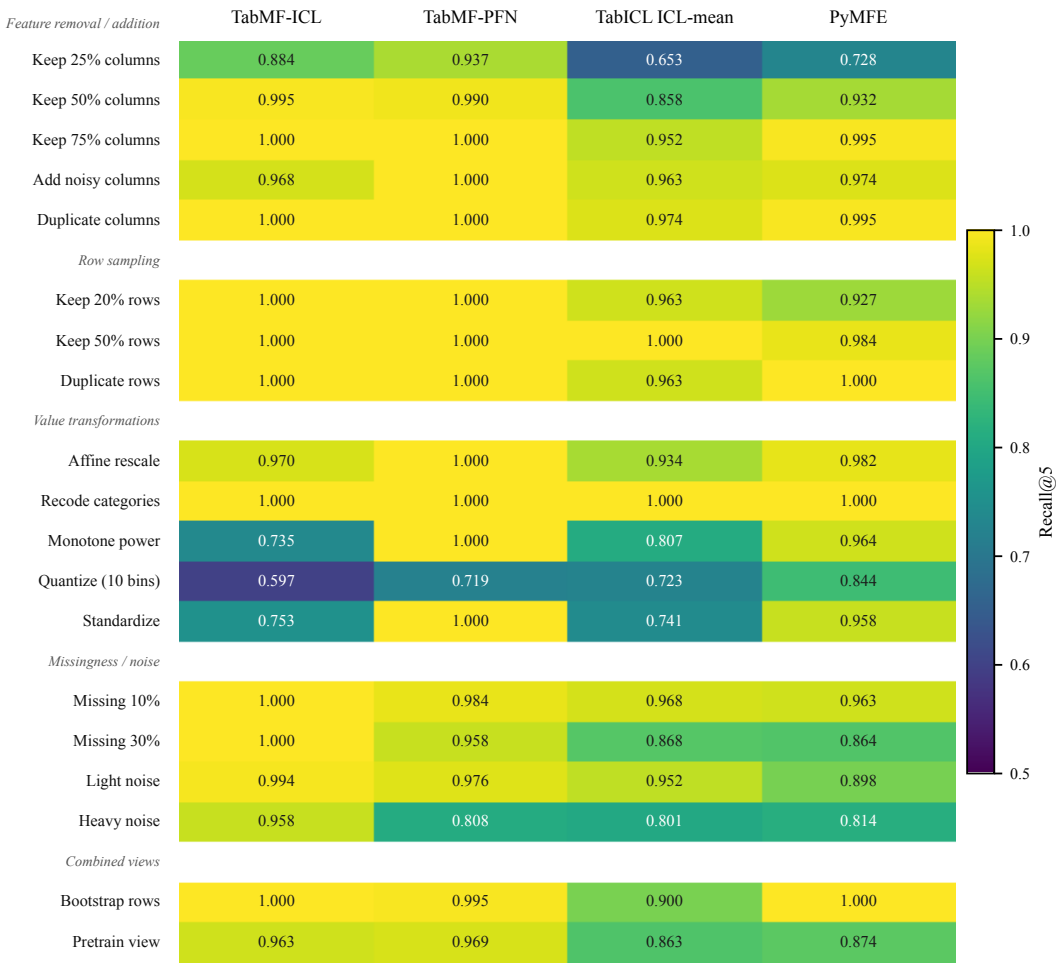


Figure 3: Retrieval recall@5 under all 19 dataset transformations for TabMF-ICL, TabMF-PFN, frozen TabICL ICL-mean, and PyMFE. The headline retrieval score averages recall@5 equally across these transformations. Dataset-level paired comparisons are reported in Table 6.

Table 7: Compact benchmark selection on TabRepo at $k = 3$. Entries are mean classifier-ranking Spearman correlations over 200 shared candidate-pool trials. Encoders are ordered by K-means performance. Bold marks numerical leaders.

Encoder	K-means	Farthest-first
TabMF-PFN	**0.9292**	**0.8229**
TabMF-ICL	0.8542	0.8146
TabPFN mean-pooling	0.8500	0.6730
Random	0.7766	0.7766
Neural Statistician	0.738	0.782
PyMFE	0.7093	0.7812
Dataset2Vec	0.6515	0.7261
TabICL ICL-mean	0.5750	0.7108

Frozen TabPFN comparison. On TabRepo, the mean paired improvements of TabMF-PFN over frozen TabPFN mean-pooling are 0.0795 with K-means (95% bootstrap interval $[0.068, 0.091]$) and 0.1500 with farthest-first selection ($[0.133, 0.167]$). Under the 12-comparison Holm family defined separately for each selector, the corresponding adjusted Wilcoxon p-values are 2.03×10^{-24} and 1.96×10^{-26}. TabMF-ICL does not significantly exceed frozen TabPFN under K-means (Holm-

Table 8: K-means benchmark-selection comparisons on TabRepo at a three-dataset budget. Differences are TABMF minus baseline in mean Spearman correlation. Two-sided Wilcoxon signed-rank tests pair the 200 candidate-pool trials. Reported p-values are Holm-adjusted jointly over the 12 comparisons.

Variant	Baseline	$\Delta\rho$	Holm p
TABMF-ICL	Random	$+0.0777$	5.28×10^{-7}
TABMF-ICL	ICL-mean	$+0.2792$	6.03×10^{-28}
TABMF-ICL	PyMFE	$+0.1450$	2.67×10^{-22}
TABMF-ICL	Dataset2Vec	$+0.2028$	1.74×10^{-25}
TABMF-ICL	Neural Statistician	$+0.1162$	2.00×10^{-19}
TABMF-ICL	Frozen TabPFN	$+0.0045$	0.2408
TABMF-PFN	Random	$+0.1526$	2.18×10^{-23}
TABMF-PFN	ICL-mean	$+0.3542$	2.07×10^{-32}
TABMF-PFN	PyMFE	$+0.2199$	7.27×10^{-32}
TABMF-PFN	Dataset2Vec	$+0.2777$	2.51×10^{-32}
TABMF-PFN	Neural Statistician	$+0.1911$	2.32×10^{-31}
TABMF-PFN	Frozen TabPFN	$+0.0795$	2.03×10^{-24}

Table 9: Sensitivity of three-dataset selection on TabRepo to K-means initialization. Entries are paired mean-correlation differences (TABMF minus baseline). Candidate pools and embeddings are fixed. The alternative initialization preserves the direction of all six comparisons.

Variant	Baseline	Original	Alternative
TABMF-ICL	ICL-mean	$+0.2792$	$+0.2602$
TABMF-ICL	PyMFE	$+0.1450$	$+0.1280$
TABMF-ICL	Dataset2Vec	$+0.2028$	$+0.1946$
TABMF-PFN	ICL-mean	$+0.3542$	$+0.3478$
TABMF-PFN	PyMFE	$+0.2199$	$+0.2155$
TABMF-PFN	Dataset2Vec	$+0.2777$	$+0.2822$

adjusted $p = 0.2408$), but does under farthest-first selection ($p = 1.62 \times 10^{-25}$). All of these encoders have complete coverage of the evaluated TabRepo population.

Robustness checks. An alternative K-means initialization preserves the direction of all six comparisons between the adapted encoders and TabICL ICL-mean, PyMFE, and Dataset2Vec (Table 9). Across $k \in \{2, 3, 4\}$, a TABMF variant obtains the highest numerical score under both K-means and farthest-first selection. The strongest variant depends on the selection budget: TABMF-ICL leads K-means at $k = 2$, whereas TABMF-PFN leads at $k = 3$ and $k = 4$ and under farthest-first selection at all three budgets (Table 10).

Coverage. HyTrel embeds 129 of the 144 TabRepo datasets. Every shared candidate pool contains at least one dataset without a HyTrel embedding, so HyTrel has no score under the common-population protocol. The Neural Statistician covers all 144 datasets; its results under both selection rules are reported in Table 7.

B.4 DATASET-PROPERTY PROBES

B.4.1 PROBE PROTOCOL

Encoders receive complete tables and their designated predictive targets when supported; regression targets are quantile-binned into at most ten classes where required by the encoder. Property labels and scores are used only by the downstream probes. We use tuned linear probes for the primary evaluation because they provide a common and interpretable measure of how readily a property can be recovered from each representation.

Table 10: TabRepo benchmark selection with two to four selected datasets. Entries are mean Spearman correlations with the full-population classifier ranking. Bold marks the best score within each fixed budget and selector. Various budgets assess sensitivity to the selection budget.

k	Selector	TabMF-ICL	TabMF-PFN	ICL-mean	PyMFE	Random	D2V	Frozen PFN	Neural Stat.
2	K-means	**0.8568**	0.6728	0.3507	0.6778	0.7209	0.6947	0.8214	0.7828
2	Farthest-first	0.7003	**0.8752**	0.5941	0.6979	0.7209	0.7100	0.6439	0.6418
3	K-means	0.8542	**0.9292**	0.5750	0.7093	0.7766	0.6515	0.8497	0.7380
3	Farthest-first	0.8146	**0.8229**	0.7108	0.7812	0.7766	0.7261	0.6728	0.7819
4	K-means	0.8504	**0.9393**	0.6743	0.8894	0.8171	0.7282	0.8536	0.7260
4	Farthest-first	0.8127	**0.8697**	0.8148	0.8338	0.8171	0.7805	0.7447	0.8560

For these probes, robust median/IQR scaling is followed by clipping to ± 8 and PCA retaining 99.9% of variance. Preprocessing is refitted within the inner folds of a grid search selecting among 17 logistic-regression penalties or 40 ridge penalties. Classification uses stratified folds and log-loss model selection; regression uses ordinary folds and an SVD ridge solver. The three-fold outer evaluation is repeated five times. PyMFE missing values are median-imputed over the evaluation population before cross-validation; the other checked embeddings contain no missing values.

Domain and paired property analyses use a common population of 190 datasets for the main encoder comparisons. Available-case regression coverage is 190 or 191 depending on the encoder, while HyTrel property probes cover 177 datasets. The primary aggregate includes encoders with reported scores on all six ranking units; individual probe analyses use their stated available-case populations.

B.4.2 Domain Discrimination

The TabBench datasets are assigned to eight broad domains: Healthcare, Behavioral, Computer Vision, Industry & Science, Other, Finance/insurance, Games & Synthetic, and Social/Public. The primary endpoint uses the tuned logistic-regression probe above and averages fold-level macro one-vs-rest AUCs within repetitions.

A separate paired dataset-bootstrap analysis uses the common 190-dataset population and the mean of five pooled-out-of-fold AUCs. Its point estimates are 0.697 for TabMF-ICL, 0.681 for TabMF-PFN, 0.661 for TabICL ICL-mean, 0.653 for TabPFN mean-pooling, and 0.641 for Dataset2Vec. None of the ten pairwise comparisons remains significant after Holm correction. This analysis applies to the pooled-out-of-fold endpoint and is therefore not attached to the fold-averaged domain scores in Table 1.

Strengthened nonlinear probe. To assess whether the linear comparison is sensitive to readout capacity, we repeat domain discrimination with a tuned SVM that jointly searches RBF and polynomial kernels after the same fold-local preprocessing. Table 11 reports all 11 encoders in the restricted comparison pool. Each encoder's summary metric uses its maximal available population; pairwise tests use the population shared by the corresponding pair. Significance is evaluated with paired Wilcoxon tests on per-dataset log-loss and Holm correction over all 55 encoder pairs. Under this stronger probe, TabMF-ICL outperforms frozen TabICL ICL-mean (AUC 0.6997 versus 0.6441; adjusted $p = 0.0278$), whereas TabMF-PFN is not significantly separated from frozen TabPFN mean-pooling.

B.4.3 Epiplexity Prediction

We assign each dataset a prequential epiplexity score motivated by learnable information for computationally bounded observers (Finzi et al., 2026). The estimate is specific to the observer and protocol described here and should not be interpreted as model-independent prediction difficulty. Categorical variables are integer-coded, missing values are median-imputed, and features are z-scored. Rows are shuffled once with seed 42 and processed in blocks of 32. An online SGD classifier with logistic loss, an adaptive learning-rate schedule, and initial learning rate $\eta_0 = 0.01$ predicts each block before updating on it. The epiplexity meter uses baseline entropy $\ln C$, corresponding to a uniform distribution over the C classes, and returns the final dataset-level score.

Table 11: Domain discrimination with the strengthened nonlinear probe. A tuned SVM jointly searches RBF and polynomial kernels. AUC and log-loss are computed on each encoder's maximal available population. Pairwise inference uses per-dataset log-loss on the population shared by each pair, with Holm correction over all 55 encoder comparisons. Bold denotes the best point estimate.

Encoder	AUC ↑	Log-loss ↓	N
TabMF-ICL	**0.6997**	**1.7860**	190
TabMF-PFN	0.6808	1.8100	191
PyMFE	0.6662	1.8871	191
TabICL row-mean	0.6637	1.8850	190
TabPFN mean-pooling	0.6620	1.8535	191
TabICL ICL-mean	0.6441	1.8585	190
Dataset2Vec	0.6239	1.9286	191
HyTrel	0.6158	1.9798	177
Neural Statistician	0.5517	2.0017	191
TabICL column-mean	0.5442	2.0313	190
TabICL random initialization	0.5256	2.0322	190

Table 12: Epiplexity prediction on a common population of 190 datasets. **A:** Mean Spearman correlation across five cross-validation repetitions. **B:** Frozen-versus-adapted comparisons from a paired dataset bootstrap, with Holm adjustment over all ten pairwise comparisons among the five encoders. Bold p-values support higher correlation for the frozen readout.

A. Common-population scores

Encoder	Spearman ρ
TabICL ICL-mean	0.764
TabPFN mean-pooling	0.760
TabMF-ICL	0.639
TabMF-PFN	0.621
Dataset2Vec	0.529

B. Frozen-versus-adapted comparisons

Frozen readout	Adapted encoder	Holm p
TabICL ICL-mean	TabMF-ICL	**0.040**
TabICL ICL-mean	TabMF-PFN	**0.040**
TabPFN mean-pooling	TabMF-ICL	0.079
TabPFN mean-pooling	TabMF-PFN	**0.040**

Primary linear-probe inference. For paired comparisons, five encoders are evaluated on the same 190 datasets, ordered by dataset identifier and using shared cross-validation partitions. Within each of five repetitions, Spearman correlation is computed over pooled out-of-fold predictions from the tuned ridge probe. We jointly resample dataset identifiers in 10,000 paired bootstrap draws; correlations are computed separately for each repetition and averaged within each draw. Holm correction is applied across all ten pairwise encoder comparisons. These tests apply to Table 12, whereas Table 1 reports available-case scores using each encoder's native coverage. Cross-validation repetitions measure sensitivity to the fold partition and are not treated as independent datasets.

On the common population, TabICL ICL-mean significantly outperforms both adapted encoders, and TabPFN mean-pooling significantly outperforms TabMF-PFN. The remaining frozen-versus-adapted comparison is not significant after correction. The two frozen encoders are not significantly separated from one another, nor are the two adapted encoders.

Strengthened nonlinear probe. We repeat epiplexity prediction with kernel ridge regression and a substantially wider joint search over RBF, Laplacian, and polynomial kernels after the same fold-local preprocessing. Table 13 reports the full 11-encoder restricted pool, using the leakage-controlled PyMFE representation. The primary metric for this capacity check is mean per-dataset absolute error, which is also the unit used by the paired Wilcoxon tests; Spearman correlation is reported for comparability with the primary evaluation. As above, summary metrics use each en-

Table 13: Epiplexity prediction with the strengthened nonlinear probe. Kernel ridge regression jointly searches RBF, Laplacian, and polynomial kernels. Mean absolute error is the endpoint used for ranking and paired inference; Spearman correlation is included for comparison with the primary linear probe. Pairwise tests use the population shared by each pair, with Holm correction over all 55 comparisons. Bold denotes the best point estimate.

Encoder	Mean $\lvert\text{error}\rvert \downarrow$	Spearman $\uparrow$	N
TabPFN mean-pooling	**0.1770**	**0.8292 ± 0.026**	191
TabMF-PFN	0.1794	0.7633 ± 0.019	191
TabICL ICL-mean	0.2032	0.7991 ± 0.021	190
TabMF-ICL	0.2224	0.6953 ± 0.012	190
Dataset2Vec	0.2265	0.6182 ± 0.026	191
Neural Statistician	0.3297	0.2658 ± 0.027	191
HyTrel	0.3348	0.1746 ± 0.038	177
TabICL column-mean	0.3447	0.2501 ± 0.043	190
TabICL row-mean	0.3465	0.2289 ± 0.092	190
TabICL random initialization	0.3511	0.1898 ± 0.052	190
PyMFE (leakage-controlled)	0.3703	0.1160 ± 0.036	191

Table 14: Explicit exclusions in the leakage-controlled PyMFE difficulty configuration. Landmarking and model-based groups are omitted entirely. The remaining general, statistical, and information-theoretic groups are fitted without the dataset target. All summaries of each listed base feature are excluded together.

Excluded base-feature identifiers		
`nr_class`	`freq_class`	`class_ent`
`class_conc`	`mut_inf`	`joint_ent`
`eq_num_attr`	`ns_ratio`	`can_cor`
`gravity`	`lh_trace`	`p_trace`
`roy_root`	`w_lambda`	`sd_ratio`

coder's maximal population and pairwise tests use the population shared by each pair, with Holm correction over all 55 comparisons. Under this stronger readout, neither TabMF variant is significantly separated from its corresponding frozen representation. Thus, the primary ridge results establish differences in linear accessibility, but do not establish that epiplexity-related information is absent from the adapted embeddings.

B.4.4 PyMFE Control for Epiplexity

Full PyMFE includes general, statistical, information-theoretic, model-based, and landmarking descriptors fitted to $(\mathbf{X}, \mathbf{y})$. For epiplexity prediction, we instead use only the general, statistical, and information-theoretic groups, fit the extractor without $\mathbf{y}$, omit the model-based and landmarking groups, and remove all summaries of the target-dependent base features listed in Table 14. This controls direct overlap with ingredients of the epiplexity score, including its class-count-dependent baseline. Full PyMFE is retained for the other evaluation tasks.

B.5 Aggregate Ranking and Statistical Scope

The primary aggregate combines six evaluation units: real-data bootstrap-view discrimination AUC, domain macro-OVR AUC, continuous epiplexity Spearman correlation, mean transformation-retrieval recall@5, TabRepo three-dataset K-means ranking Spearman, and TableShift shift-severity Spearman correlation. Each unit ranks the same seven encoders with complete task coverage, using average ranks for ties. Synthetic model-selection scores, farthest-first selection, web-table transfer, and hyperparameter transfer do not enter the aggregate. No significance test is applied to the aggregate ranks.

We assess sensitivity to the benchmark-selection task by varying the K-means selection budget over $k \in \{2, 3, 4\}$ and by omitting the task entirely. All comparisons retain the same seven encoders

Table 15: Sensitivity of the six-task aggregate ranking to the TabRepo benchmark-selection budget. Only the TabRepo K-means evaluation changes across columns; the other five evaluation units are held fixed. The aggregate score is the median rank across the six evaluation units.

Encoder	TR $k=2$	TR $k=3$	TR $k=4$	Agg. $k=2$	Agg. $k=3$	Agg. $k=4$
TABMF-PFN	6	1	1	2.0	**1.5**	**1.5**
TABMF-ICL	1	2	4	**1.5**	2.0	2.0
TabICL ICL-mean	7	7	7	3.5	3.5	3.5
PyMFE	5	5	2	4.0	4.0	3.5
Frozen TabPFN	2	3	3	5.0	5.0	5.0
Dataset2Vec	4	6	5	5.0	5.5	5.0
Neural Statistician	3	4	6	7.0	7.0	7.0

Table 16: TableShift tasks used for the shift-severity evaluation. The accuracy gap is the mean ID-minus-OOD accuracy difference across estimators with both published ID and OOD results. Row counts are the actual ID and OOD samples embedded after the at-most-8,192-row sampling described in Section 4.

Task	Dataset ID	Gap	ID rows	OOD rows
ASSISTments	`assistments`	0.3519	8192	1906
FICO HELOC	`heloc`	0.2826	278	6914
College Scorecard	`college_scorecard`	0.1132	8192	1352
Hypertension	`brfss_blood_pressure`	0.0816	8192	8192
Hospital Readmission	`diabetes_readmission`	0.0489	4287	8192
Diabetes	`brfss_diabetes`	0.0450	8192	8192
Food Stamps	`acsfoodstamps`	0.0261	8192	8192
Income	`acsincome`	0.0221	8192	8192
Unemployment	`acsunemployment`	0.0193	8192	8192

and leave the other five evaluation units unchanged (Table 15). Both TABMF variants retain lower median ranks than every baseline in all four settings. Their relative ordering depends on the selection budget: TABMF-ICL leads at $k = 2$, whereas TABMF-PFN leads at $k = 3$ and $k = 4$. Without benchmark selection, both obtain a median rank of 2.0.

Bootstrap hard-margin and benchmark-selection tests use the pairing units and correction families specified in their respective subsections. Probe bootstraps resample dataset identifiers while retaining the repeated predictions associated with each sampled identifier. They characterize variation over the evaluated dataset population conditional on the fitted predictions and do not include retraining within each bootstrap draw. TableShift contains nine task-level observations; we therefore report the observed correlations without asserting pairwise encoder superiority.

TableShift shift-severity inputs. Table 16 reports the nine task-level inputs used for the shift-severity evaluation. For each task, the severity target is the mean ID accuracy minus the mean OOD accuracy over the TableShift estimators for which both published values are available. Thus, each task contributes one severity value to the Spearman correlation; no estimator-level observations are treated as independent samples.

B.6 ENCODER VARIANTS

Table 2 compares retrieval performance across initialization schemes, training objectives, and readouts. For the initialization comparison, architecture, readout, objective, and batch regime are held fixed within each backbone. Pretrained initialization outperforms scratch initialization for both TabICL and TabPFN.

For the remaining TabPFN variants, the pretrained column-aware encoder with NT-Xent obtains 0.965 recall, compared with 0.935 for the summary-token encoder and 0.957 for the pseudo-test encoder.

Table 17: Supplementary transfer evaluations. TEmBed MAP covers 16 datalakes except PyMFE (†: 17); shuffling covers 60 slices. Standard deviations are across datalakes or slices. Similarity and shuffling use uncapped inputs. HPO covers 82 targets except HyTrel (‡: 77). A dash denotes an unavailable result.

Encoder	MAP ↑	Shuffling ↑	Type F1 ↑
TABMF-ICL	0.605 ± 0.158	0.732 ± 0.281	0.363
TABMF-PFN	0.448 ± 0.117	0.773 ± 0.251	0.289
TabICL ICL-mean	0.468 ± 0.137	0.753 ± 0.266	—
HyTrel	0.684 ± 0.146	0.977 ± 0.053	0.681
PyMFE	$0.456 \pm 0.137^{\dagger}$	0.705 ± 0.150	0.372
Encoder	**HPO quality ↑**	**HPO regret@5 ↓**	**Targets**
TABMF-ICL	0.468	0.037	82
TABMF-PFN	0.676	0.032	82
TabICL ICL-mean	0.773	0.032	82
HyTrel	$0.664^{\ddagger}$	$0.042^{\ddagger}$	77

B.7 WEB-TABLE AND HYPERPARAMETER TRANSFER

Table 17 reports web-table and hyperparameter-transfer evaluations separately from the primary suite.

For web-table transfer, TEmBed similarity averages MAP over 16 GitTables datalakes (Hulsebos et al., 2023) for the TFM encoders and HyTrel, while the PyMFE aggregate covers 17 datalakes and is marked accordingly. Shuffling covers 60 slices. Reported standard deviations measure variation across datalakes or slices rather than confidence intervals. Similarity and shuffling use uncapped table inputs. These evaluations concern target-agnostic table representations and therefore differ from the supervised-dataset representation setting considered in the main evaluation.

For hyperparameter transfer, aggregates cover 82 targets across 16 search spaces for the adapted encoders and TabICL ICL-mean, compared with 77 targets for HyTrel. Within each search space, embeddings are median-imputed and standardized, and each target retrieves its single Euclidean-nearest meta-training dataset ($m = 1$). We transfer the five configurations ranked highest on that source dataset and report the minimum normalized target regret; neighbor quality normalizes the corresponding transfer performance between a random-source reference (zero) and an oracle-source reference (one). Aggregates are pooled across spaces by target count.

B.8 ENCODER EFFICIENCY AND SCALING

We additionally characterize the computational cost of producing dataset representations as table size increases. We vary the number of observations from 500 to 200,000 while fixing the number of features to 64, and vary the number of features from 8 to 2,048 while fixing the number of observations to 10,000. Each setting is evaluated on three independently generated synthetic datasets.

Figure 4 shows embedding time and peak memory. The main observation is that both TABMF variants closely track the computational cost of their corresponding frozen TFM representations: TABMF-ICL follows TabICL ICL-mean, while TABMF-PFN follows TabPFN mean-pooling. Thus, the dataset-level adaptation and readout introduce little additional inference cost beyond their respective underlying backbones. Because Dataset2Vec and the Neural Statistician subsample a bounded input regardless of table size, their constant memory and low embedding time reflect that cap rather than cheaper processing of the full table; TabMF, like its backbone, encodes the entire input.

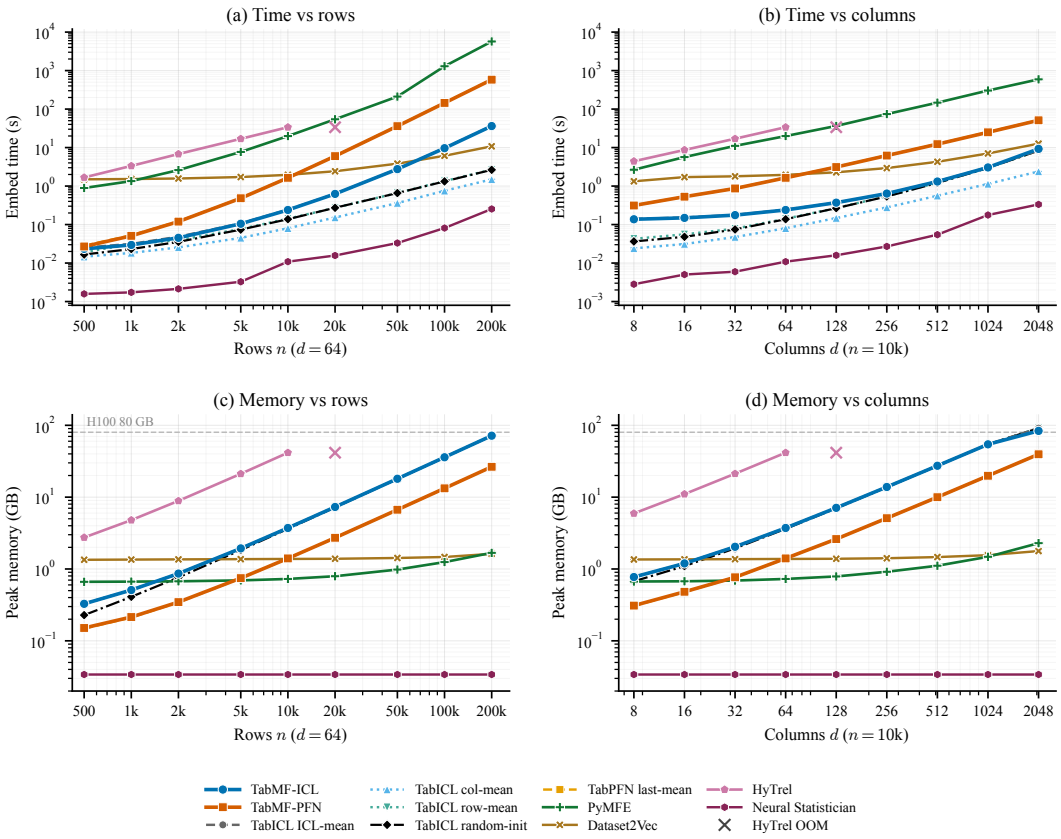


Figure 4: Encoder time and memory scaling as the number of rows and features increases. For the row-scaling experiment, the number of features is fixed at $d = 64$; for the feature-scaling experiment, the number of observations is fixed at $n = 10{,}000$. TabMF-ICL closely tracks TabICL ICL-mean, while TabMF-PFN closely tracks TabPFN mean-pooling, indicating that the dataset-level readouts introduce little additional inference cost relative to their respective backbones. HyTrel exceeds available memory at larger table sizes. Neural Statistician and Dataset2Vec use bounded input sampling, which limits their scaling with table size.

C ILLUSTRATION OF THE TABMF BACKBONE AGGREGATIONS

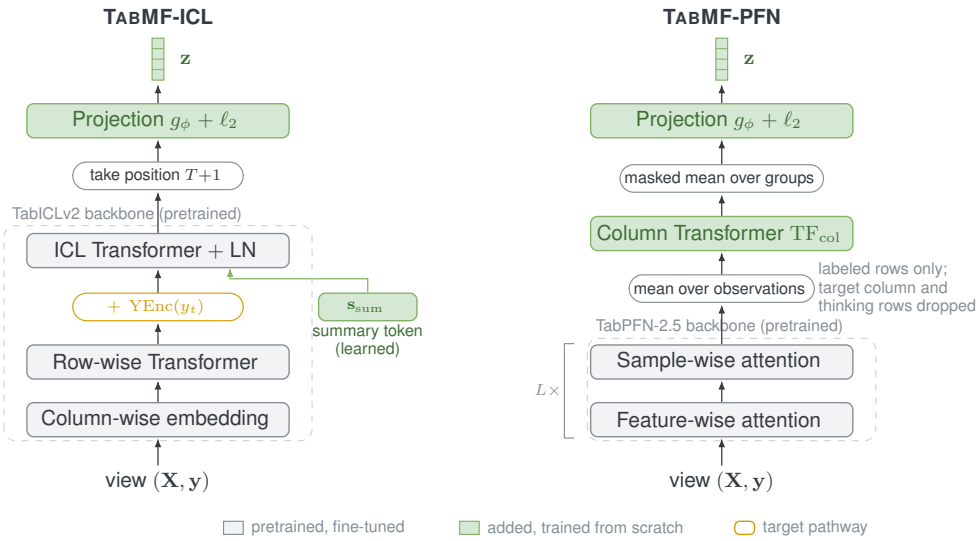


Figure 5: The two TABMF encoders. TABMF-ICL (left) appends a learned summary token to the target-conditioned row states and reads its state after the pretrained ICL Transformer and final normalization. TABMF-PFN (right) pools the backbone's feature-group states over labeled observations, models their interactions with a learned column Transformer, and averages the active groups. Each readout feeds a projection network; gray blocks are pretrained and fine-tuned, while green blocks are added and trained from scratch.