
Do Tabular Foundation Models Know Physics? Contamination, Units, and the Deterministic Limit

Wassim Tenachi

Département de physique, Université de Montréal
Mila – Quebec Artificial Intelligence Institute
wassim.tenachi@mila.quebec

Yashar Hezaveh

Département de physique, Université de Montréal
Mila – Quebec Artificial Intelligence Institute
yashar.hezaveh@mila.quebec

Laurence Perreault Levasseur

Département de physique, Université de Montréal
Mila – Quebec Artificial Intelligence Institute
levassel@mila.quebec

Pierre-Luc Bacon

Department of Computer Science and Operations Research, Université de Montréal
Mila – Quebec Artificial Intelligence Institute
pierre-luc.bacon@mila.quebec

Abstract

Tabular foundation models (TFMs) learn to fill in tables the way language models fill in text — and tables are arguably the format in which most physical measurement arrives. Did they learn any physics in the process? They are Bayesian by construction, so the question is what their prior contains. We probe it directly, evaluating four of them (TabPFN-3, TabICLv2, TabDPT and Real-TabPFN-2.5) against six baselines on datasets sampled from 316 physical equations, in and out of domain. TFMs dominate, out of the box and after tuning. But they struggle to represent noiseless deterministic mechanisms in their uncertainty, and show no sign of exploiting physical units: they interpolate physics without yet being able to act as physical models.

1 Introduction

Nature, in Galileo’s phrase, is written in the language of mathematics [Galilei, 1623]. We do not read it directly, however: we observe it as data, and in the physical sciences that data is overwhelmingly tabular [Ochsenbein et al., 2000]. Large language models have made striking progress on the mathematical language itself [Alon et al., 2026], and vision models on scientific imaging [Parker et al., 2026]. Tables, arguably the format in which most physical measurement actually arrives, have received far less attention. Skimming oceans of text produced powerful representations of the physical world [Team et al., 2023]. Would skimming oceans of tables do the same?

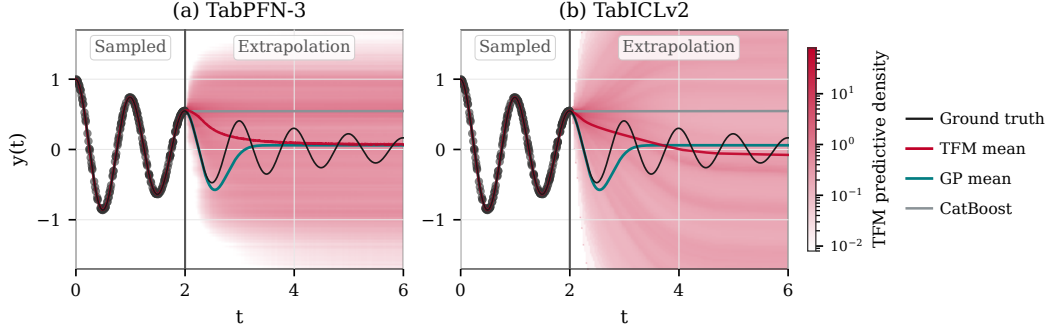


Figure 1: **TFMs interpolate physics but do not learn enough of it to extrapolate.** A noiseless damped harmonic oscillator queried beyond the sampled range, against baselines. The oscillation is recovered where there is data and lost immediately outside it, though the TFM means still settle near the equilibrium the oscillator decays to, rather than holding the last observed value. Details and other models in App. A.

Tabular foundation models (TFMs) learn to fill in tables much as language models learn to fill in text: entries are masked and the model is trained to recover them from context [Hollmann et al., 2022]. They are invariant to row and column permutations, and most emit a full predictive distribution per cell. A new task is therefore solved in a single forward pass — in-context learning (ICL), with no gradient step — so fitting is free, amortized by pretraining, while querying costs grow with context size. Most are pretrained exclusively on synthetic structural causal models (SCMs)¹ [Hollmann et al., 2022, Qu et al., 2025]; others on corpora of real tables [Ma et al., 2024, Garg et al., 2025].

Related work. TFMs are now applied across scientific fields [Grinsztajn et al., 2026], including astrophysics [Valença et al., 2026], though almost exclusively in interpolation. Benchmarking efforts such as TabArena [Erickson et al., 2026] and BeyondArena [Purucker et al., 2026] have concentrated on stochastic processes, and have pushed the field toward extrapolation, where TFMs remain weak — strikingly so on physical signals (Fig. 1). But how should a model extrapolate an arbitrary real-world table without having learned something of the physics that generated it?

Two obstacles. Physics-equation data sits close enough to the ecosystem these models are trained from² to warrant a contamination audit before interpreting any result (§3). Yet the physics regime is itself absent from the literature: BeyondArena drops deterministic-function data as artificial, and no current TFM has been pretrained on a physical law with units or a continuous target (App. E).

Which prior? TFMs are Bayesian by construction: pretraining on samples from a prior under a proper scoring rule makes the forward pass an approximation to the posterior predictive [Hollmann et al., 2025]. They return posteriors, not point estimates. The question is what prior those posteriors are built on, and whether it has anything to do with physics — and data sampled from known physical laws is the cleanest way to ask.

2 Protocol

Data. We evaluate on three sets of data sampled from physical equations: Feynman (120 equations) [Udrescu and Tegmark, 2020], standardized in SRBench [La Cava et al., 2021], and LSR-Transform (111) and LSR-Synth (85) [Shojaee et al., 2025], which rewrite Feynman into unusual forms and compose novel terms respectively. Every table is regenerated from its equation rather than loaded, so no published table is reused and n_{samples} , noise and sampling domain stay free.

Regimes. Training points are drawn from each equation’s sampling box, test points come either from the same box ($k=1$, in-domain) or from a shell outside it at twice its half-width ($k=2$, extrapolation). Targets carry multiplicative noise $y' = y(1 + \varepsilon)$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, emulating the relative error of

¹Random graphs in which each node is generated from its parents plus independent noise.

²Feynman-equation datasets make up ~ 118 of the 1053 datasets surveyed as candidates for TabArena, mostly via PMLB [Olson et al., 2017]; the final 51-dataset suite excludes them.

Table 1: **Evaluated tabular foundation models.** *SCMs*: pretrained on synthetic tables drawn from structural causal models. *Real data*: pretrained on real-world tables. *Reg.*: those real tables include regression targets, as opposed to classification only. Only TabDPT has been pretrained on real-world regression data.

Model	Output	SCMs	Real data	Reg.	Ref.
TabPFN-3	Quantiles	✓	×	—	[Grinsztajn et al., 2026]
TabICLv2	Quantiles	✓	×	—	[Qu et al., 2025]
Real-TabPFN-2.5	Quantiles	✓	✓	×	[Garg et al., 2025]
TabDPT	Point	×	✓	✓	[Ma et al., 2024]

physical instruments. We sweep $n_{\text{samples}} \in \{50, 200, 2000\}$, $\sigma \in \{0, 0.01, 0.1\}$, $k \in \{1, 2\}$ and 3 seeds, with the test set fixed at 2000 points throughout. The detailed protocol is given in App. C.

Baselines and evaluation. Following BeyondArena [Purucker et al., 2026], we evaluate the four open-weight TFMs (Table 1) against six trained baselines: RealMLP, TabM, a vanilla MLP, CatBoost, a Gaussian process (GP) and Ridge (App. B). We score with $\text{NMSE} = \mathbb{E}[(y - \hat{y})^2] / \mathbb{E}[(y - \bar{y})^2]$; since raw error spans orders of magnitude across equations, we rank models within each (task, regime) cell and report mean rank per stratum.

Fairness. Comparing a forward pass to a training loop requires fixing a budget, so we report both. (i) Under *default*, every method uses library defaults. (ii) Under *tuned*, the trained baselines get 25 random-search configurations, optionally combined by post-hoc greedy ensemble selection³; TFMs have no hyperparameters, and their analogue is the ensemble size $n_{\text{estimators}}$, which averages the forward pass over input permutations. We report both families at their cheapest and strongest settings.

3 Results & Analysis

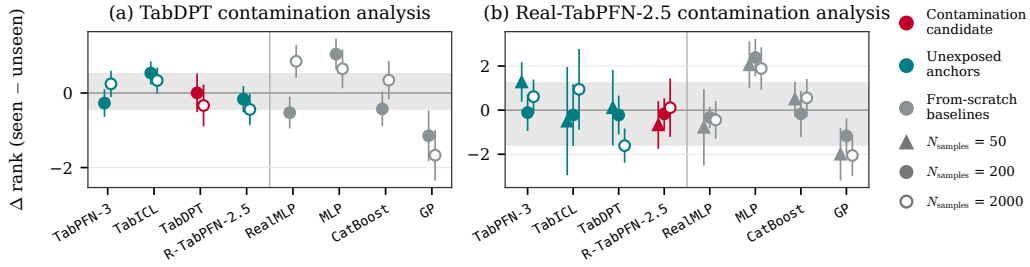


Figure 2: **No detectable contamination from real-data pretraining.** Difference in mean rank between datasets seen during pretraining and comparable unseen datasets, for (a) TabDPT and (b) Real-TabPFN-2.5. Negative values indicate better performance on the seen trained data. The shaded band spans the unexposed TFM anchors; both candidates fall inside it at every context size.

No detectable contamination. Before asking whether TFMs know physics, we must rule out that they have simply seen our evaluation data. We audited the pretraining corpora of the two real-data models, both of which enumerate them: TabDPT’s 122 datasets and Real-TabPFN-2.5’s 43. Neither contains a Feynman equation, nor any regression target generated by an analytic function. Physics appears only as detector classification, never as law⁴ (full corpora in App. E).

A list of names is not a list of generating processes, however: both corpora were filtered against benchmark suites by dataset name and content hash, and a table resampled from the same equation under a different name would pass both checks. We therefore measure exposure rather than assert it,

³Caruana-style [Caruana et al., 2004] forward selection with replacement over a model’s own 25 configurations, scored on validation; no additional fits are needed.

⁴MAGIC, HIGGS and MiniBooNE on OpenML.

comparing each model on datasets drawn from its own corpus against comparable datasets outside it, and reading the gap relative to models that saw neither⁵ (Fig. 2).

Both come back null: each candidate sits inside the unexposed band at every context size, and what little movement exists runs counter to the mechanistic prediction, growing with context size where prior exposure should matter most when context is thin. Most tellingly, models with no pretraining at all show *larger* seen-versus-unseen gaps than the exposed models do, so this statistic is dominated by which datasets fall in each pool rather than by exposure to them. That spread also bounds what we could have detected: any contamination effect below roughly one rank position would be invisible at this pool size.

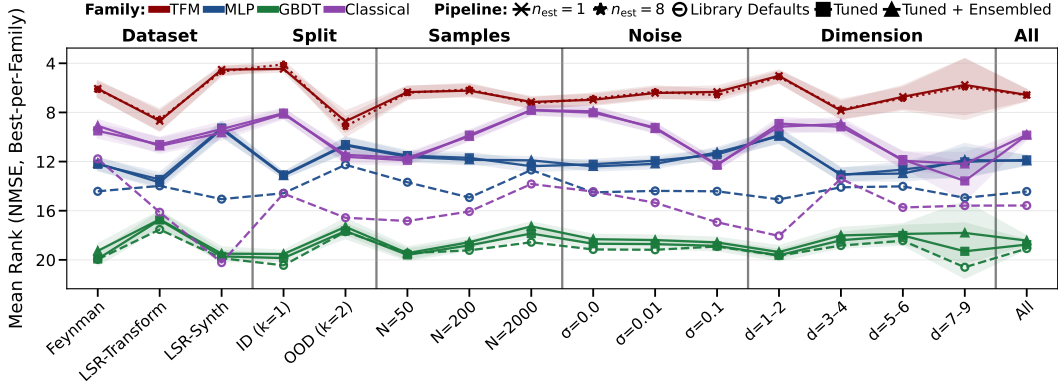


Figure 3: **The TFM lead on physics data survives tuning and ensembling.** Mean rank within each stratum (lower is better). Details and per-model results in App. D — TabPFN-3 leads throughout.

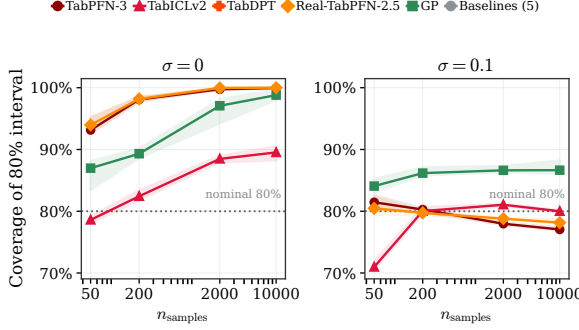
Tuning does not close the gap. Under default configurations our ordering reproduces BeyondArena’s: TFMs lead the trained baselines across every stratum (Fig. 3). Their reported reversal arises from tuning, which we therefore evaluate directly. It does not occur here. Per model (Fig. 6), TabPFN-3 still leads under all three pipelines in every stratum, the sole exception being the Gaussian process at the largest context size, and only against the TFM’s cheapest setting. It also holds where both conditions that favor the trained models apply at once — out of domain at the largest context size — though the margin narrows there⁶. The clearest breakdown in ordering comes not from tuning but from extrapolation: out of domain the field compresses and the ranking reshuffles more than anywhere else (Fig. 6), models converging on a common failure rather than degrading in place.

Can the prior represent a noiseless mechanism? TFMs are pretrained on synthetic data in which every variable carries independent noise, how do they behave on noiseless data? On data drawn from a deterministic law f , aleatoric noise is zero, so in interpolation the correct posterior predictive approaches a delta function as context grows. At finite context its width is not zero, since several functions still fit the observed points, but it should shrink as the predictions improve. A well-calibrated model’s 80% interval should therefore keep covering 80% of test points as n_{samples} grows.

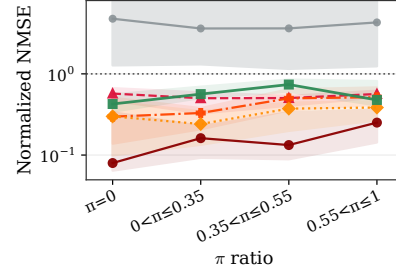
In the noisy regime ($\sigma=0.1$), the TFMs stay close to 80% (Fig. 4a-right). In the noiseless regime ($\sigma=0$), their coverage instead climbs toward 100% (Fig. 4a-left): the intervals shrink more slowly than the error, so the models become underconfident, as if a residual noise floor kept their predictive distributions too wide. The effect is strongest for the two TabPFN models, near 100% from $n_{\text{samples}}=200$, and mild for TabICLv2. The GP shows it later, as expected: its fitted noise level has a built-in lower bound, a noise floor by construction. A prior placing noise on every mechanism would explain this, but TabICLv2 is also pretrained on SCMs and is less affected, so SCM pretraining alone may not account for the effect; the way intervals are produced (ensembling, preprocessing) may also contribute.

⁵For TabDPT, the 28 regression datasets in its corpus against 37 matched datasets from TabArena and OpenML-CTR23. For Real-TabPFN-2.5, the single function-generated table it has seen (fried, and only in binarized form) against friedman2 [Olson et al., 2017], an unexposed member of the same family — one dataset, hence the noisier panel.

⁶Mean rank over 314 tasks: 8.47 for TabPFN-3 against 9.85 for the tuned and ensembled GP, the best baseline here.



(a) **TFMs are calibrated with noise but underconfident without it.** Fraction of in-domain test points inside each model’s 80% predictive interval (median over tasks), a calibrated model stays on the dotted line. At $\sigma=0.1$ the TFM’s sit near it beyond the smallest context, at $\sigma=0$ their coverage climbs toward 100% as context grows, most strongly for the two TabPFN models.



(b) **Dimensional structure leaves little trace on error.** Median NMSE relative to the typical model on each task (below 1 is better), binned by the fraction of input variables left as dimensionless groups, n_π/n_{vars} ; the $n_\pi=0$ tasks are all pure power laws. Other models vary less than TabPFN-3, whose lead is largest on the most reducible tasks.

Does the prior exploit dimensional structure? Physical variables carry units, and by Buckingham’s theorem a law of n variables whose units involve k independent dimensions reduces, up to a prefactor fixed by the units, to a function of only $n - k$ dimensionless groups. A physical problem is thus often lower-dimensional than its column count. A TFM sees only numbers and has no direct access to this. We count the dimensionless groups n_π each task’s inputs can form⁷ and ask whether a model’s error follows the raw column count or n_π . For every model it follows the column count; once that is accounted for, n_π adds little for most. The clearest exception is TabPFN-3, whose lead over the other models is largest on the most reducible tasks ($n_\pi=0$) and shrinks as n_π/n_{vars} grows (Fig. 4b). All 26 tasks with $n_\pi=0$ are pure power laws, which dimensional analysis fixes up to a constant, so this more likely reflects their simpler form than any use of units. We find no evidence that any model exploits dimensional structure.

4 Conclusion

Tabular foundation models interpolate physical data well, without ever having been pretrained on a physical law, and we find no detectable contamination behind this. Their limits lie elsewhere: they become underconfident on noiseless data and show no sign of exploiting dimensional structure — two properties every physical law has and no random structural causal model does. They are excellent amortized interpolators, but a physical model would need a prior that contains physics. Pretraining on continuous physical targets — absent from every corpus we audited — would be a natural way to find out, as would comparing them with time-series foundation models on dynamical systems.

Acknowledgments

This work is partially supported by Schmidt Futures, a philanthropic initiative founded by Eric and Wendy Schmidt as part of the Virtual Institute for Astrophysics (VIA). The work is in part supported by computational resources provided by Calcul Québec and the Digital Research Alliance of Canada. Y.H. and L.P. acknowledge support from the Canada Research Chairs Program, the Natural Sciences and Engineering Research Council of Canada through grants RGPIN-2020-05073 and 05102.

References

Noga Alon, Thomas F Bloom, W Timothy Gowers, Daniel Litt, Will Sawin, Arul Shankar, Jacob Tsimmerman, Victor Wang, and Melanie Matchett Wood. Remarks on the disproof of the unit distance conjecture. *arXiv preprint arXiv:2605.20695*, 2026.

⁷ $n_\pi = n_{\text{vars}} - \text{rank}(D)$, with D the matrix of SI dimension exponents of the input variables, from the Feynman units table.

- Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18, 2004.
- 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. *Advances in Neural Information Processing Systems*, 38, 2026.
- Galileo Galilei. *Il saggiaiore*. 1623.
- 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 preprint arXiv:2507.03971*, 2025.
- Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.
- Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. Tabm: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations*, volume 2025, pages 77899–77935, 2025.
- Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, et al. Tabpfn-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.
- Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.
- Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.
- Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeyer, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.
- David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned mlps and boosted trees on tabular data. In *Neural Information Processing Systems*, 2024.
- William La Cava, Bogdan Burlacu, Marco Virgolin, Michael Kommenda, Patryk Orzechowski, Fabrício Olivetti de França, Ying Jin, and Jason H Moore. Contemporary symbolic regression methods and their relative performance. *Advances in neural information processing systems*, 2021 (DB1):1, 2021.
- Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L Caterini, and Maksims Volkovs. Tabdpt: Scaling tabular foundation models on real data. *arXiv preprint arXiv:2410.18164*, 2024.
- Calvin McCarter. What exactly has tabpfn learned to do? In *ICLR Blogposts 2024*, 2024. URL <https://iclr-blogposts.github.io/2024/blog/what-exactly-has-tabpfn-learned-to-do/>.
- François Ochsenbein, Patricia Bauer, and James Marcout. The vizier database of astronomical catalogues. *Astronomy and Astrophysics Supplement Series*, 143(1):23–32, 2000.
- Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(1):36, Dec 2017. ISSN 1756-0381. doi: 10.1186/s13040-017-0154-4. URL <https://doi.org/10.1186/s13040-017-0154-4>.

- Liam Parker, Francois Lanusse, Jeff Shen, Ollie Liu, Tom Hehir, Leopoldo Sarra, Lucas Meyer, Micah Bowles, Sebastian Wagner-Carena, Helen Qu, et al. Aion-1: Omnimodal foundation model for astronomical sciences. *Advances in Neural Information Processing Systems*, 38:95386–95428, 2026.
- Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.
- Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David Holzmüller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, Gaël Varoquaux, and Frank Hutter. Beyond iid: How general are tabular foundation models, really?, 2026. URL <https://arxiv.org/abs/2606.30410>.
- Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.
- Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.
- Parshin Shojaee, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D Doan, and Chandan K Reddy. Llm-srbench: A new benchmark for scientific equation discovery with large language models. *arXiv preprint arXiv:2504.10415*, 2025.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science advances*, 6(16):eaay2631, 2020.
- Raquel R Valença, Lilianne Nakazono, Rafael Izbicki, Marco Henrique de Almeida Inácio, Maycon Jorge Deláqua da Silva, Kiana Coimbra Buin Lins, Natanael M Cardoso, Claudia Mendes de Oliveira, et al. Tabular foundation models for the estimation of probabilistic quasar photometric redshifts in s-plus. *arXiv preprint arXiv:2608.10280*, 2026.

A Extrapolating a damped harmonic oscillator

We qualitatively assess how TFMs extrapolate physical dynamics by asking them to continue a damped harmonic oscillator, a canonical physical system, and comparing their predictions with those of a Gaussian process (GP) and CatBoost. Two are pretrained on synthetic SCMs only (TabPFN-3 [Grinsztajn et al., 2026], TabICLv2 [Qu et al., 2025]), one on real tables only (TabDPT [Ma et al., 2024]), and one on both (Real-TabPFN-2.5 [Garg et al., 2025]). We use $n = 200$ samples and no noise, so that any predictive width or extrapolation failure reflects the model rather than the data. Results are shown in Fig. 5.

Extrapolation. All four models fail to continue the oscillation past the sampled range, relaxing to a constant within roughly half a period. They do settle near the equilibrium: TabPFN-3 and TabICLv2 both decay toward zero, as the true signal does, rather than holding the last observed value as CatBoost does. This is consistent with McCarter [2024], who found no evidence that TabPFN detects periodic structure — and our TabPFN panels are indeed a smooth, phase-agnostic blur.

Two exceptions, and a caveat. TabICLv2’s predictive density shows wave-like banding past the boundary: the model hedges across a range of values rather than collapsing to one, and the resulting ridges echo the scale of the oscillation. TabDPT continues the signal for roughly a quarter period before flattening. Both postdate McCarter [2024] and both are suggestive. However, a phase cross-correlation of each model’s density against the true continuation shows that the banding is *not* phase-locked⁸. The structure is real multimodality, not a recovered frequency. We report it as an encouraging direction rather than as periodic extrapolation.

A floor even where the answer is known. Inside the sampled range the data are noiseless and densely sampled, so the correct predictive distribution is close to a point mass. The GP’s interval is effectively zero there (about 10^{-5} , normalized); unlike the default GP of §3, this one allows its noise level down to 10^{-12} , and the optimizer drives it there. All three distributional TFMs instead report a small but nonzero width (0.010–0.020), the same residual floor as in §3. TabDPT appears to escape this, but only because it emits no predictive distribution at all (Table 1) — it does not report zero uncertainty, it reports none.

⁸We slide the true continuation in time by 201 lags spanning one period and measure how much of each model’s predictive density falls along each shifted curve. Had a model learned the oscillation’s phase, the unshifted curve — the correct answer — would score best. It does not, ranking 98th of 201 for TabICLv2 and 157th for TabPFN-3. TabDPT emits no density and cannot be tested this way.

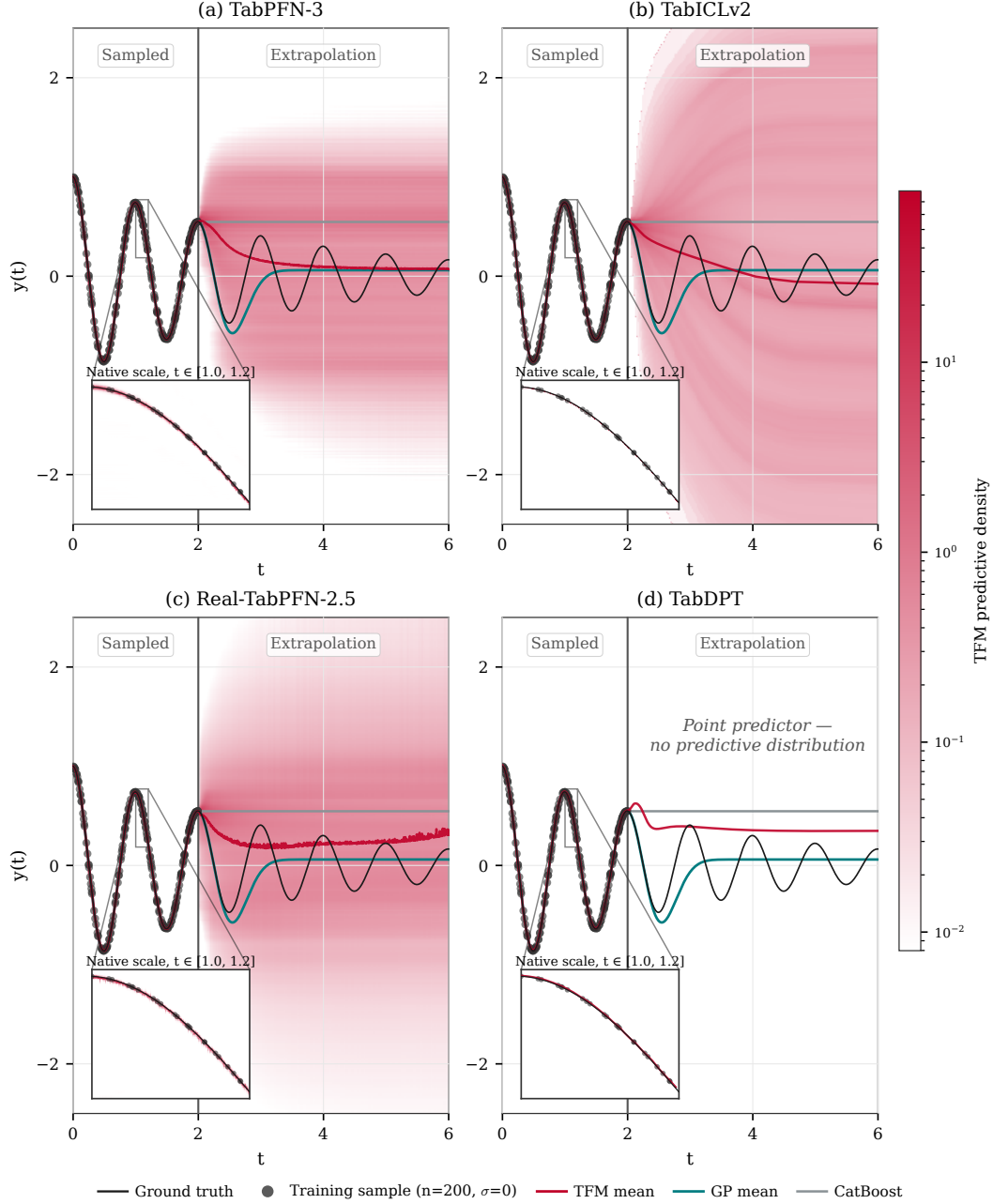


Figure 5: **Predictive distributions of all four TFMs on a damped harmonic oscillator.** Same setup as Fig. 1: $n = 200$ noiseless samples over $t \in [0, 2]$ (points), queried to $t = 6$, against a Gaussian process (GP) and CatBoost. Color is the TFM predictive density; insets zoom an in-domain window at native scale. Inside the densely sampled range the correct interval is close to a point: the GP’s is effectively zero ($\sim 10^{-5}$, normalized), yet all three distributional TFMs report 0.010 (TabICLv2) to 0.019–0.020 (the two TabPFN variants) — the uncertainty floor of §3, seen directly. Outside it, every model relaxes to a constant: TabICLv2’s density retains an oscillatory banding, TabDPT shows a partial oscillation before flattening, and CatBoost holds the last observed value. TabDPT emits no predictive distribution (§2) and is shown as a mean only.

B Trained baselines

Trained baselines. Table 2 lists the six methods fitted per task, spanning the deep, tree-based and classical families. Of these, only the Gaussian process emits a predictive distribution; the rest return point estimates and therefore cannot appear in any uncertainty comparison. The GP uses an RBF kernel plus a learned white-noise term at scikit-learn defaults, with targets normalized; its noise level is bounded below at 10^{-5} of the target variance, so its 80% interval cannot fall below about $0.008 \text{std}(y)$ — the noise floor discussed in §3.

Table 2: **Trained baselines.** Each is fitted per task. Predictive output type is determined empirically from each model’s implementation rather than from its documentation; only the Gaussian process returns a closed-form predictive density.

Model	Family	Output	Ref.
RealMLP	MLP with tuned defaults	Point	[Holzmüller et al., 2024]
TabM	MLP ensemble	Point	[Gorishniy et al., 2025]
MLP	Vanilla MLP	Point	[Gorishniy et al., 2021]
CatBoost	Gradient-boosted trees	Point	[Prokhorenkova et al., 2018]
GP	Gaussian process	Density	[Seeger, 2004]
Ridge	Linear, ℓ_2	Point	[Hoerl and Kennard, 1970]

C Protocol details

The extrapolation shell. Axis j of a task’s sampling box has range $[l_j, h_j]$, center c_j and half-width w_j . Test points at scale k are drawn from $[c_j - k w_j, c_j + k w_j]$ with rejection of any point falling inside the $k=1$ box, so the test set is a shell rather than a superset of the training domain — without rejection, most of an OOD draw would land in-domain and the split would measure interpolation. All d axes are extended simultaneously. We use per-axis box extension rather than a convex-hull criterion: the latter is exponential in d , and these domains are boxes by construction. The OOD region is defined from the training box alone, and it, the noise model and the failure rule were all fixed before any result was inspected.

Seeds in place of cross-validation. TabArena and BeyondArena evaluate over repeated CV folds. Because we generate data rather than load it, the equivalent is a fresh draw per seed: each seed independently resamples both training and test points, so no CV is needed. We use three seeds and take the median, which is robust to a single catastrophic draw. TFMs require multiple seeds too — nothing is fitted, but ensemble permutations and internal preprocessing vary between runs.

Failure policy. A run that runs out of memory, times out or returns NaN is recorded as failed rather than dropped, and imputed at NMSE=1 in aggregates — equivalently $R^2=0$, i.e. predicting the training mean. This keeps a failure interpretable and model-independent rather than letting it silently vanish from a mean. Across the full campaign, 12 of 362,231 runs failed permanently — all TabDPT, and all tracing to a single numerical edge case on one task at $n_{\text{samples}}=2000$, repeated across noise levels, splits and protocols.

Validation carve-out. Under the tuned and ensembled pipelines, the trained baselines carve 20% of n_{samples} for model selection and so fit on $0.8 n_{\text{samples}}$; TFMs condition on all of it, since they have nothing to select. We disclose this rather than equalize it: capping the TFM’s context to match would change the quantity being measured, and giving the baselines a separate validation set would change the sample budget. The gap is largest at $n_{\text{samples}}=50$, where the trained models see 40 rows against the TFM’s 50, and is the most likely place for a tuned-versus-default comparison to understate the benefit of tuning.

Preprocessing. Column order is randomized per seed for every model. TFMs are built to be column-permutation-invariant; checking this empirically costs nothing and appears as a variance term. We do not disable each model’s internal preprocessing (TabPFN applies its own quantile and power transforms) and record which was active in every result record: this is a confound to disclose, not to remove. Metrics are always computed in the original target space.

Uncertainty evaluation. Fig. 4a uses the Feynman set in domain ($k=1$), extended with $n_{\text{samples}}=10,000$ on a 40-task subset at $\sigma \in \{0, 0.1\}$. Coverage is the fraction of test points inside the central 80% predictive interval (10th to 90th percentiles), computed per task, then the median is taken over seeds and over tasks.

Cost accounting. Wall-clock is recorded per run and split into fit and predict phases: a TFM is nearly free to fit and costly to query, a trained model the reverse, and only the split makes that visible. We do not attempt FLOPs, which are not well defined across transformer, tree and kernel families and ignore memory movement. Hardware is recorded rather than pinned, so timing comparisons are filtered at analysis time; accuracy results are hardware-independent. Pretraining cost is excluded by convention, as for ImageNet-pretrained vision models.

Reproducibility. Each run is keyed by a hash of its resolved configuration and skipped if the record already exists, making the campaign idempotent and resumable. Every record embeds the full configuration, the git commit, key package versions, hardware identifiers and per-phase timings. The main grid comprises $316 \text{ tasks} \times 3 n_{\text{samples}} \times 3 \text{ noise levels} \times 2 \text{ split scales} \times 3 \text{ seeds} \times 10 \text{ models}$ under each protocol; the tuned protocol multiplies this by 25 configurations per trained baseline. Total measured cost was ≈ 1300 GPU-hours (H100 and A100) and ≈ 4200 CPU-hours.

D Per-model benchmark results

Figure 3 collapses each family to its best-performing member under each pipeline, following BeyondArena’s presentation. Figure 6 unpacks the same ranks into all 26 model–pipeline entities. This appendix records what the collapse hides, and several observations for which there was no room in the main text.

How the ranks are computed. Within each (task, regime) cell we rank all 26 entities jointly — the four TFMs at $n_{\text{est}} \in \{1, 8\}$ and the six trained baselines at library defaults, tuned, and tuned with post-hoc ensembling — then average over tasks within a stratum. Ranks therefore run from 1 to 26 in both figures, and are not comparable to a ranking computed over a smaller pool. Cells falling below 80% task coverage are dropped rather than plotted.

Which model wins each family. The collapse in Fig. 3 is stable for three families of four: TabPFN-3 is the best TFM in every stratum under both settings, GP beats Ridge everywhere, and CatBoost is the only gradient-boosted entrant. The MLP family is the exception. TabM is the strongest tuned member overall, but ReaMLP takes over at library defaults out of domain and at the smallest context size, and again once ensembling is applied at the largest. Statements about "the MLP family" in the main text should be read accordingly.

Ensembling contributes almost nothing. Post-hoc greedy selection improves only ReaMLP by a measurable margin; for TabM, CatBoost, GP and Ridge the median change is indistinguishable from zero, with selection typically returning the single already-best configuration. This is not a consequence of an under-sized selection budget: allowing 25, 50, 100 or 200 rounds with replacement over the same library yields results identical to four decimal places, and selection saturates at four to ten distinct configurations in every case. We attribute this to the data rather than the protocol. Ensemble selection exploits diversity in the errors made by different configurations, and on noiseless deterministic targets differently-tuned fits of the same model converge on the same function and therefore make correlated errors. That ReaMLP is the exception is consistent with this account: its search space varies architecture and training schedule widely enough to produce genuinely different fits.

Where tuning does help. Tuning improves every trained baseline more at the largest context size than at the smallest, as expected, and improves every model except Ridge at $\sigma = 0$, where library defaults — calibrated for noisy real-world tables — are furthest from appropriate.

A note on TabDPT. TabDPT separates from the other TFMs specifically on LSR-Synth and on low-dimensional tasks, where it performs worse than its synthetic-prior counterparts. This is not evidence of contamination — the effect runs in the wrong direction, since LSR-Synth is the cleanest of our three sets — but it is a real difference between models pretrained on real corpora and those pretrained on synthetic ones, and we record it here without an explanation.

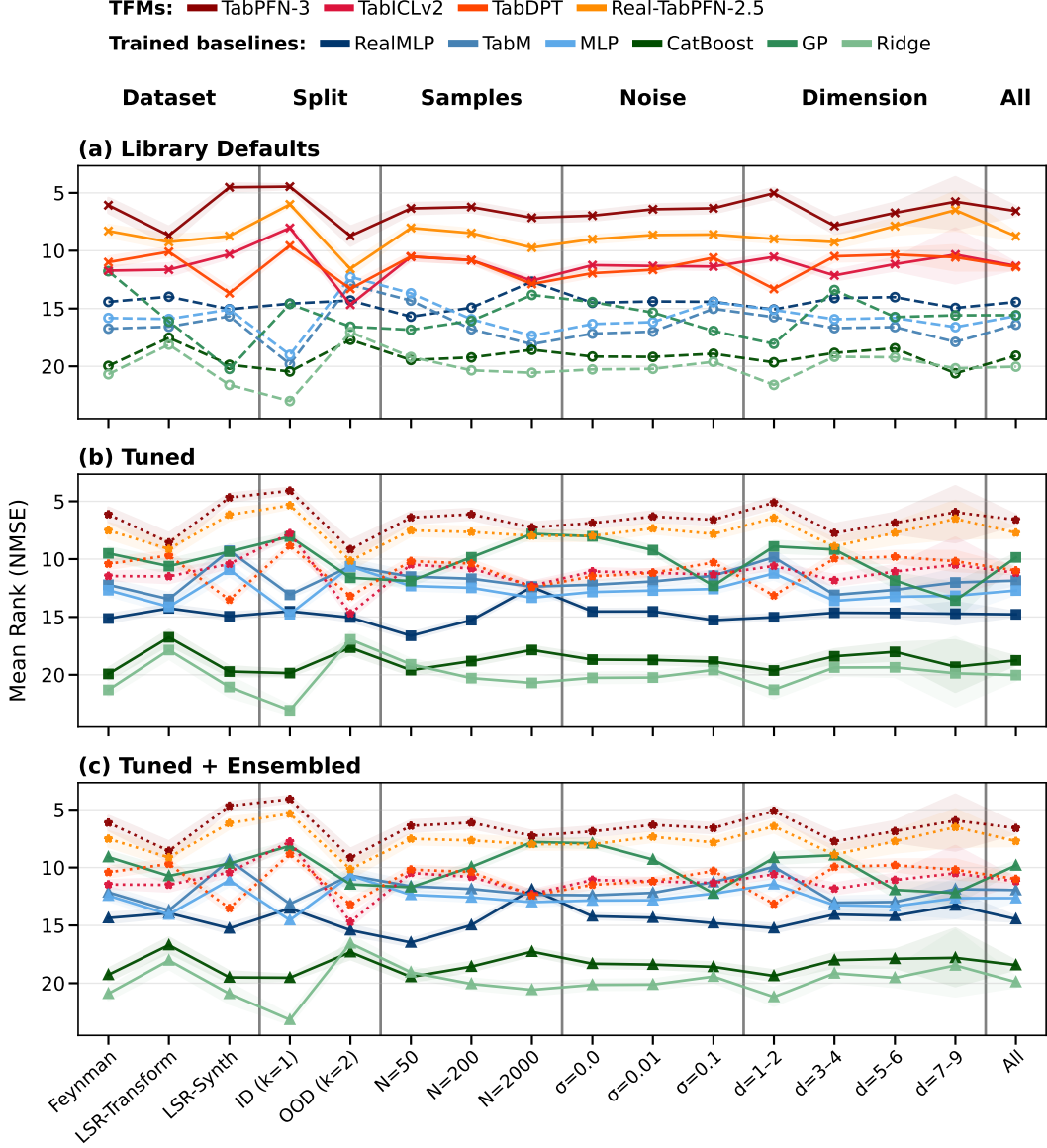


Figure 6: **Per-model results across all strata.** The same ranks as Fig. 3, unpacked into all 26 model-pipeline entities and split by pipeline: **(a)** library defaults, **(b)** tuned and **(c)** tuned with post-hoc ensembling. The six trained baselines appear in every panel; the four TFMs, which have no tuning, are repeated throughout as a fixed reference, at $n_{\text{est}}=1$ in (a) and $n_{\text{est}}=8$ in (b) and (c). Panels share a common rank axis, so vertical gaps are comparable across them. Cells falling below 80% task coverage are dropped rather than plotted.

E Pretraining corpora of the real-data TFMs

Both real-data-pretrained models enumerate their pretraining corpora, which is what makes the audit in §3 possible. We reproduce them here, grouped and with live links, so that the claims in the main text can be checked directly.

TabDPT’s corpus (Table 3)⁹ comprises 122 OpenML datasets, 93 classification and 28 regression, with domain labels assigned by its authors. All six datasets labeled *Physics/astronomy* are classification tasks — particle-collision and Cherenkov-shower event discrimination, spacecraft sensor state,

⁹Reproduced from Appendix B of Ma et al. [2024].

satellite land cover — and none of the nine labeled *Deterministic and simulated* has a continuous target. No regression target in the corpus is generated by an analytic function.

Real-TabPFN-2.5’s corpus (Table 4)¹⁰ comprises 43 datasets from OpenML and Kaggle. It contains no real-world regression table at all: its one function-generated entry, `fried`, links to OpenML 901, the binarized variant of Friedman #1 in which the continuous target has been replaced by a two-class label thresholded at its mean. Two Sloan Digital Sky Survey catalogs appear, both as star/galaxy/quasar classification.

Table 3: TabDPT **pretraining corpus** (122 datasets), grouped by the domain labels assigned in its own appendix and sorted alphabetically within group. Regression targets are marked [†]; all others are classification, except `Census-Income*`, for which no target type is given. No regression target in the corpus is generated by an analytic function, and every dataset in the *Physics/astronomy* and *Deterministic and simulated* groups is a classification task.

<i>Physics/astronomy</i> — 6 datasets, 0 regression		
<code>higgs</code>	<code>magic</code>	<code>MagicTelescope</code>
<code>MiniBooNE</code>	<code>Satellite</code>	<code>shuttle</code>
<i>Deterministic and simulated</i> — 9 datasets, 0 regression		
<code>artificial-characters</code>	<code>chess</code>	<code>hill-valley</code>
<code>kr-vs-k</code>	<code>kropt</code>	<code>parity5_plus_5</code>
<code>poker-hand</code>	<code>puma8NH</code>	<code>twonorm</code>
<i>Other science</i> — 4 datasets, 1 regression		
<code>gas-drift</code>	<code>gas-drift-different-concentrations</code>	<code>musk</code>
<code>sulfur</code> [†]		
<i>Biology/ecology</i> — 11 datasets, 1 regression		
<code>cjs</code>	<code>covertype</code>	<code>GAMETES_Epistasis_2-Way_20atts</code>
<code>mushroom</code>	<code>one-hundred-plants-texture</code>	<code>OVA_Endometrium</code>
<code>OVA_Lung</code>	<code>pollen</code>	<code>sylva_agnostic</code>
<code>visualizing_soil</code> [†]	<code>yeast</code>	
<i>Medical/human sensor</i> — 11 datasets, 4 regression		
<code>cardiotocography</code>	<code>Diabetes130US</code>	<code>eeg-eye-state</code>
<code>eye_movements</code>	<code>ldpa</code>	<code>pbcseq</code>
<code>QSAR-TID-10980</code> [†]	<code>QSAR-TID-11</code> [†]	<code>topo_2_1</code> [†]
<code>walking-activity</code>	<code>yprop_4_1</code> [†]	
<i>Financial/demographic</i> — 15 datasets, 2 regression		
<code>ada_agnostic</code>	<code>ada_prior</code>	<code>Census-Income*</code>
<code>credit</code>	<code>default-of-credit-card-clients</code>	<code>heloc</code>
<code>house_16H</code>	<code>house_8L</code>	<code>house_prices_nominal</code> [†]
<code>ipums_la_97-small</code>	<code>ipums_la_98-small</code>	<code>ipums_la_99-small</code>
<code>kdd_internet_usage</code>	<code>nursery</code>	<code>us_crime</code> [†]
<i>Human behaviour</i> — 17 datasets, 8 regression		
<code>Bike_Sharing_Demand</code> [†]	<code>black_friday</code> [†]	<code>Buzzinsocialmedia_Twitter</code> [†]
<code>Click_prediction_small</code>	<code>compas-two-years</code>	<code>KDDCup09-Upselling</code>
<code>KDDCup09_appetency</code>	<code>nyc-taxi-green-dec-2016</code> [†]	<code>okcupid-stem</code>
<code>OnlineNewsPopularity</code> [†]	<code>porto-seguro</code>	<code>road-safety</code>
<code>Santander_transaction_value</code> [†]	<code>seattlecrime6</code> [†]	<code>sf-police-incidents</code>
<code>SpeedDating</code>	<code>wine_quality</code> [†]	
<i>Industrial/operational</i> — 10 datasets, 4 regression		
<code>airlines</code>	<code>Airlines_DepDelay_10M</code> [†]	<code>Allstate_Claims_Severity</code> [†]
<code>Amazon_employee_access</code>	<code>APSFailure</code>	<code>delays_zurich_transport</code> [†]
<code>KDDCup09_churn</code>	<code>kick</code>	<code>Mercedes-Benz_Greener_Manufacturing</code> [†]
<code>pol</code>		
<i>Computing</i> — 4 datasets, 3 regression		
<code>KDDCup99</code>	<code>MIP-2016-Reg.</code> [†]	<code>SAT11-HAND-runtime-Reg.</code> [†]
<code>SGEMM_GPU_kernel_performance</code> [†]		
<i>Vision/audio/text features</i> — 7 datasets, 0 regression		
<code>amazon-commerce-reviews</code>	<code>fbis_wc</code>	<code>JapaneseVowels</code>
<code>lals_wc</code>	<code>page-blocks</code>	<code>scene</code>
<code>spoken-arabic-digit</code>		
<i>Other or not provided</i> — 28 datasets, 5 regression		
<code>ada</code>	<code>Ailerons</code> [†]	<code>albert</code>
<code>analcatadata_halloffame</code>	<code>analcatadata_supreme</code> [†]	<code>christine</code>
<code>colleges</code> [†]	<code>colleges_usnews</code>	<code>dilbert</code>
<code>dionis</code>	<code>elevators</code>	<code>fabert</code>
<code>guillermo</code>	<code>helena</code>	<code>IMDB_drama</code>
<code>jannis</code>	<code>jasmine</code>	<code>madeline</code>
<code>mofn-3-7-10</code>	<code>particulate-matter-ukair-2017</code> [†]	<code>philippine</code>
<code>PieChart3</code>	<code>PizzaCutter3</code>	<code>riccardo</code>
<code>robert</code>	<code>sylvine</code>	<code>volkert</code>
<code>Yolanda</code> [†]		

¹⁰Reproduced from Appendix C.1 of Grinsztajn et al. [2025].

Table 4: Real-TabPFN-2.5 **pretraining corpus** (43 datasets), as listed in its technical report, sorted alphabetically within source. Every entry is a classification task: the corpus contains no real-world regression table. Its one function-generated entry, `fried`, links to OpenML 901 — the binarised variant of Friedman #1, in which the continuous target has been replaced by a two-class label thresholded at its mean.

<i>OpenML</i> — 20 datasets		
<code>artificial-characters</code>	<code>BNG(breast-w)</code>	<code>BNG(tic-tac-toe)</code>
<code>connect_4</code>	<code>eeg-eye-state</code>	<code>Employee-Turnover-at-TECHCO</code>
<code>eye_movements</code>	<code>FOREX_eurpln-hour-High</code>	<code>fried</code>
<code>gas-drift</code>	<code>higgs</code>	<code>Internet Firewall Data</code>
<code>Intersectional-Bias-Assessment-(Training-Data)</code>	<code>school-admission-binary</code>	<code>Medical-Appointment</code>
<code>microaggregation2</code>	<code>mushroom</code>	<code>NewspaperChurn</code>
<code>nursery</code>	<code>WBCAtt</code>	
<i>Kaggle</i> — 23 datasets		
<code>aam_avaliacao_dataset</code>	<code>Air Traffic Data</code>	<code>ansible-defects-prediction</code>
<code>AV Healthcare Analytics II</code>	<code>Candidate Selection</code>	<code>Cardio Disease</code>
<code>Classification - Crop Damages in India (2015-2019)</code>	<code>CSGO Round Winner Classification</code>	<code>Flower Type Prediction Machine Hack</code>
<code>Horse Racing - Tipster Bets</code>	<code>How severe the accident could be</code>	<code>hr-comma-sep</code>
<code>ip-network-traffic-flows-labeled-with-87-attacks</code>	<code>ipnetwork cross-sell prediction</code>	<code>L&T Vehicle Loan Default Prediction</code>
<code>League of Legends Diamond Games (First 15 Minutes)</code>	<code>Richter's Predictor Modeling Earthquake Damage</code>	<code>Server Logs - Suspicious</code>
<code>Sloan Digital Sky Survey DR14</code>	<code>Sloan Digital Sky Survey DR16</code>	<code>Term Deposit Prediction Data Set</code>
<code>trajectory-based-ship-classification</code>	<code>Travel Insurance</code>	