Real-TabPFN: Improving Tabular Foundation Models via Continued Pre-training With Real-World Data

Anurag Garg [1] **Muhammad Ali** [1] **Noah Hollmann** [2] **Lennart Purucker** [1] **Samuel Müller** [1 3] **Frank Hutter** [2 4 1]

Abstract

Foundation models for tabular data, like TabPFN, achieve strong performance on small datasets when pre-trained solely on synthetic data. We show that this performance can be significantly boosted by a targeted continued pre-training phase. Specifically, we demonstrate that leveraging a small, curated collection of large, real-world datasets for continued pre-training yields superior downstream predictive accuracy compared to using broader, potentially noisier corpora like CommonCrawl or GitTables. Our resulting model, Real-TabPFN, achieves substantial performance gains on 29 datasets from the OpenML AutoML Benchmark.

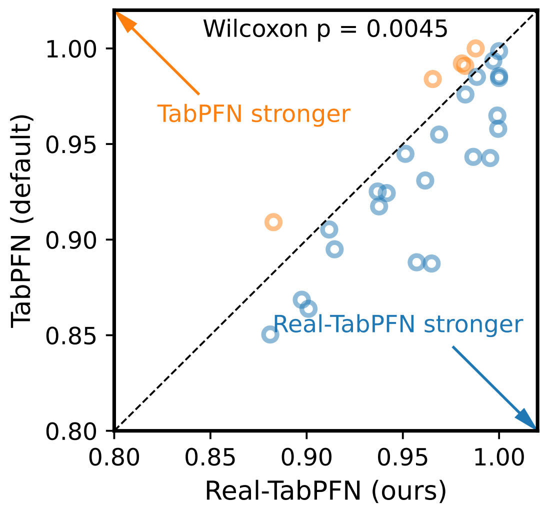

Figure 1. Per Dataset Normalized ROC Comparison of TabPFN (default) and Real-TabPFN (ours) on the 29 datasets from the OpenML AutoML Benchmark Datasets. Wilcoxon p refers to the two-sided Wilcoxon signed-rank test p value.

1. Introduction

Until recently, traditional tree-based algorithms like XGBoost (Chen & Guestrin, 2016) and CatBoost (Dorogush et al., 2017) have consistently outperformed neural networks on tabular prediction tasks (Grinsztajn et al., 2022). However, TabPFNv2 has recently demonstrated improved performance on small datasets (up to 10,000 samples and 500 features), significantly advancing deep learning for tabular data.

Although TabPFNv2 delivers strong average performance, it is still not universally best-in-class: many datasets are better handled by well-tuned tree ensembles or by careful hyper-parameter searches on TabPFNv2 itself. It is not easy to improve the TabPFNv2 model further, since it has already been exhaustively trained on over 100 million synthetic tables that approximate a broad prior over tabular problems. However, even small accuracy gains could translate into tangible benefits, such as fewer hospital re-admissions or more precise credit-risk scoring, domains in which tabular data dominates.

We enhance TabPFNv2's in-context learning by continuing to pre-train it on a carefully selected set of real-world tables from OpenML (Vanschoren et al., 2013) and Kaggle[1]. The resulting model, **Real-TabPFN**, consistently outperforms TabPFNv2 on the OpenML AutoMLBenchmark classification tasks (see Figure 1). In practice, Real-TabPFN serves as a stronger off-the-shelf baseline for tabular classification than the default TabPFNv2 model.

Our contributions are:

➢ We empirically show that using real-world datasets with synthetic data during the pre-training of a tabular foundation model boosts in-context learning performance, opening a promising research direction.

[1]Department of Informatik, University of Freiburg, Freiburg, Germany [2]Prior Labs GmbH, Freiburg, Germany [3]Meta, New York, USA (work done in at University of Freiburg) [4]ELLIS Institute Tübingen, Tübingen, Germany. Correspondence to: Anurag Garg <garga@cs.uni-freiburg.de>.

Proceedings of the 1st ICML Workshop on Foundation Models for Structured Data, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

[1]https://www.kaggle.com

➢ The new model **Real-TabPFN** and its public weights[2], an extension of TabPFNv2 obtained by continued pre-training on real-world data. Real-TabPFN's in-context learning outperforms its predecessor on 29 small tabular datasets.

2. Related Work

So far, tabular foundation models have been pre-trained solely on synthetic or real-world data. Our work aims to bridge this gap via continued pre-training.

Synthetic-only Tabular Foundation Models. TabPFN pre-trains a transformer (Vaswani et al., 2017) on millions of synthetically generated tabular datasets, achieving strong in-context learning performance on small datasets (Hollmann et al., 2023; 2025). Several extensions retain the synthetic-data recipe: TabForestPFN (den Breejen et al., 2025) augments TabPFN with more complex, decision-boundary-oriented generators, and TabICL (Qu et al., 2025) scales to tables with 500k rows.

Real-data Foundation Models. In parallel, purely real-data approaches emerged. TabDPT (Ma et al., 2024) couples retrieval-based self-supervision with discriminative transformers and is trained on real-world data collected from OpenML. TabuLa-8B (Gardner et al., 2024) adapts an Llama 3-8B backbone via language modeling over serialized tables, demonstrating that large LLMs can transfer to tabular few-shot prediction after real-world pre-training.

3. Pre-training and Evaluation Data

To study continued pre-training, we had to decide on the data for continuing pre-training and for evaluation.

Evaluation Data. We adopt the same datasets as used by Hollmann et al. (2025) for TabPFNv2 to evaluate our method. We use the same 29 datasets from the OpenML AutoML Benchmark (Gijsbers et al., 2023); see Appendix B. All datasets contain up to 10,000 samples and 500 features.

Continued Pre-training Data. Unlike the domains of natural language processing and computer vision, where many carefully curated datasets, such as ImageNet (Deng et al., 2009), COCO (Lin et al., 2015), FineWeb (Penedo et al., 2024), and C4 (Raffel et al., 2019) are available, comparably high-quality resources for tabular learning remain scarce.

Recent efforts have attempted to address this gap. Notable contributions include the Web Table Corpus (Bizer et al., 2015), TabLib (Eggert et al., 2023), and GitTables (Hulsebos et al., 2021). Beyond these, researchers either generate synthetic datasets or rely on existing repositories like UCI (Kelly et al., 2007), OpenML (Vanschoren et al., 2013), and

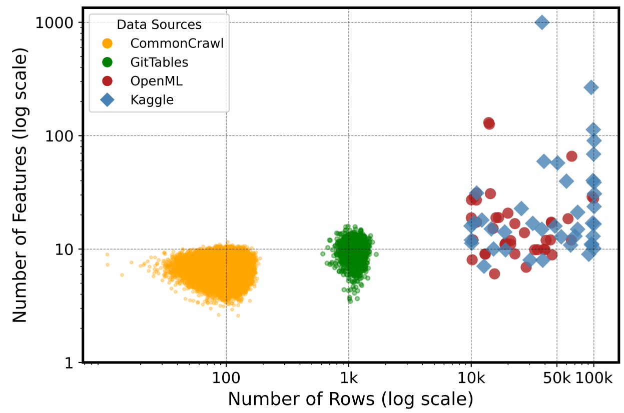

Figure 2. Distribution of dataset sizes (number of rows and features) from various sources. The prevalence of smaller datasets in broad corpora like CommonCrawl and GitTable contrasts with the larger datasets from OpenML and Kaggle.

Kaggle. We adopt the latter, manually curating 71 high-quality datasets from OpenML and Kaggle. Figure 2 shows the distribution of the number of features and rows across datasets from different sources; Appendix A lists the curated datasets used for continued pre-training. We apply minimal preprocessing to the 71 datasets: categorical features are encoded with Scikit-learn's (Pedregosa et al., 2011) OrdinalEncoder; and if the target variable has more than ten classes, we retain the nine most common classes and merge the remainder into a single tenth class.

Data Contamination. We carefully avoided data contamination (Jiang et al., 2024) between training and the evaluation set. We implemented a multi-tiered filtering process: **(1)** We only select datasets exceeding 10,000 samples since all our evaluation datasets have fewer samples. **(2)** We cross-referenced dataset IDs, names, and shapes to identify potential duplicates. **(3)** We compared feature names across datasets to detect similar or identical data structures. **(4)** We generated hashes of both rows and columns to identify potential data duplications at a granular level. **(5)** We manually inspected metadata to ensure no training and evaluation datasets share a common source dataset or a subsampled version of that dataset.
We exclude any dataset from the pre-training data that does not meet these criteria.

4. Method: Continued Pre-training of TabPFN

Our method bridges purely synthetic training (e.g., TabPFN (Hollmann et al., 2025; 2023)) and purely real-data training (e.g., TabDPT (Ma et al., 2024)) by leveraging the complementary strengths of both paradigms.

[2]available on Hugging Face

Concretely, we adopt a *two-stage approach*. **Stage 1** relies on the original TabPFNv2 checkpoint, pre-trained by Hollmann et al. (2025) on a large, diverse set of synthetic tables. This serves as our starting point. **Stage 2** continues pre-training exclusively on a curated collection of heterogeneous real-world tables. This approach contrasts with *mixed* training, where synthetic and real samples are fed to the model simultaneously, as in D'souza et al. (2025). We opted for a two-stage approach as it is easier to apply and builds directly on a strong existing synthetic base model.

Although continued pre-training has shown remarkable success in language models (Gururangan et al., 2020), its potential for tabular foundation models remains largely unexplored. By pre-training on diverse real datasets rather than narrow task-specific data, our approach improves generalization while preserving cross-domain adaptability.

To enable robust continued pre-training, we retained the original TabPFNv2 architecture and trained with a reduced learning rate of 3×10^{-7} using the AdamW optimizer (Loshchilov & Hutter, 2017a) together with a linear warm-up followed by a cosine annealing schedule (Loshchilov & Hutter, 2017b).

Moreover, we added a regularizer penalizing distance to the L2-Starting-Point (L2-SP) (Li et al., 2018) to the pre-training objective. This penalizes large deviations from the initial pre-trained weights, and is used to mitigate catastrophic forgetting (Kirkpatrick et al., 2017). More precisely, let $\mathbf{w}^0$ denote the parameter vector of the pre-trained base model from which continued pre-training begins. The L2-SP penalty then regularizes the model parameters towards this initial vector. It is formally defined as:

$$\Omega(\mathbf{w}) = \frac{\alpha}{2} \left\|\mathbf{w} - \mathbf{w}^0\right\|_2^2,$$

where α controls the strength of the regularization penalty, and $\|\cdot\|_2$ denotes the L2 norm. We add the L2 norm to the cross-entropy loss: $\mathcal{L} = \mathcal{L}_{\text{CE}} + \Omega(\mathbf{w})$ to obtain our final pre-training objective. We used a regularization strength α of 0.003.

We continued pre-training for 20,000 steps with a batch size of 1 (i.e., a single dataset). Choosing a batch size of 1 is a simple approach that naturally handles the varying feature dimensions of real-world datasets without requiring padding or truncation. Critically, it also allowed us to maximize the training context for each dataset up to 20,000 samples, limited primarily by GPU memory, rather than by batching constraints. For datasets larger than this limit, we sample uniformly up to 20,000 rows. Per batch, we split the data into 60% context (TabPFN's training data) and 40% query (TabPFN's testing data) for the forward pass. The training was performed on a single Nvidia RTX 2080 Ti GPU. To stay within GPU memory limits, we further capped

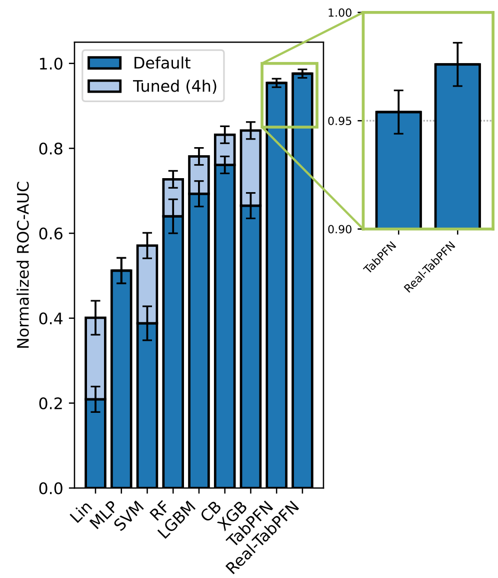

Figure 3. Mean Normalized ROC AUC Comparsion of Real-TabPFN with all the default and the tuned versions of the baselines on the AutoMLBenchmark. Scores were normalized per dataset, with 1.0 representing the best and 0.0 the worst performance with respect to all baselines.

each dataset at 400,000 total cells, adjusting the number of samples accordingly for datasets with too many attributes.

5. Experiments and Results

We follow the evaluation protocol of Hollmann et al. (2025) and evaluate Real-TabPFN with 10-fold cross-validation per dataset. Furthermore, we reuse the performance values for additional baselines from the results reported by Hollmann et al. (2025). The baselines were tuned for ROC-AUC via five-fold cross-validated random search under a four-hour time budget. We run Real-TabPFN and TabPFNv2 ourselves without hyperparameter tuning to focus on their in-context learning performance.

Figure 1 compares TabPFNv2 and Real-TabPFN per dataset. We observe that Real-TabPFN significantly outperforms TabPFNv2. Additionally, Figure 3 shows that Real-TabPFN improves the mean normalized ROC-AUC from 0.954 to 0.976 and naturally outperforms all baselines on average, like TabPFNv2. We provide a table with various additional performance metrics for all methods in Appendix C.

Effect of Context Size. We investigate the impact of the size of datasets during continued pre-training by testing pre-training with datasets from 2,048 to 20,000 (our GPU memory limit) samples. Figure 4 shows that downstream accuracy increases with a larger context size.

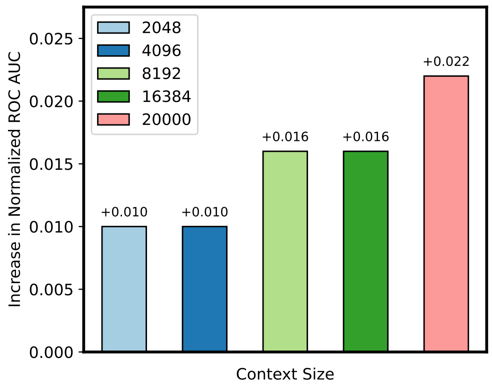

Figure 4. Increase in normalized ROC AUC as the continued-pre-training context grows. The gains are shown relative to the base TabPFNv2 model performance which was synthetically pre-trained with 2,048 context size.

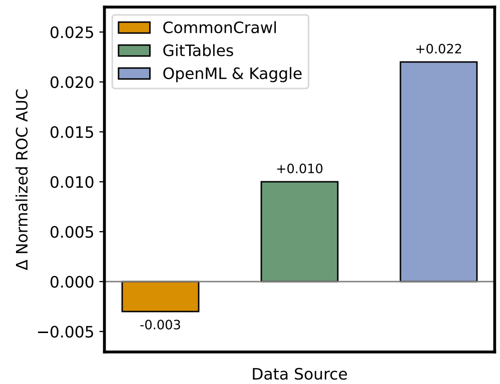

Figure 6. Change in normalized ROC AUC as the training data source is varied. The changes are shown relative to the base TabPFNv2 model performance which was synthetically pre-trained.

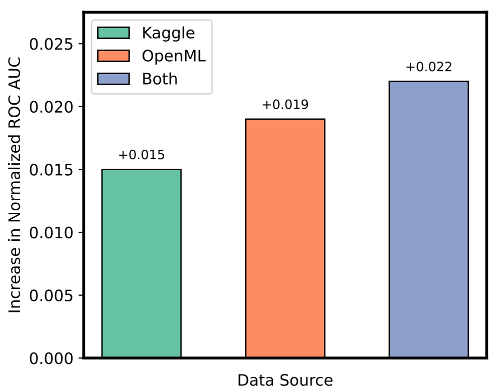

Figure 5. Increase in normalized ROC AUC as the training data source is varied. The gains are shown relative to the base TabPFNv2 model performance which was synthetically pre-trained.

OpenML vs Kaggle. To understand the impact of our final curated 71 training data on model performance, we repeated continued pre-training with three corpora: (1) only KAGGLE, (2) only OPENML, and (3) the union of both. As Figure 5 shows, OPENML alone delivers a $+0.019$ gain, while KAGGLE alone gives $+0.015$. Combining them yields the strongest boost, $+0.022$, confirming that heterogeneous sources provide complementary supervision signals. This finding indicates that while OpenML datasets provide slightly better performance individually, the combination of both data sources yields the best performing model.

Effect of Training Data Source. To evaluate the impact of different training data sources, we also experimented with two alternative corpora: (1) COMMONCRAWL (Yin et al., 2020) and (2) GITTABLES (Tran et al., 2024). We applied aggressive filtering by evaluating datasets with Logistic Regression (Cox, 1958) and Random Forest (Breiman, 2001) and subsequently removing noisy datasets, followed by our data contamination pipeline (see Section 3). This resulted in approximately $97,000$ CommonCrawl and 658 GitTables datasets.

Figure 6 compares performance using these two corpora. The model trained on CommonCrawl (approximately 100 data points and 7 features on average per dataset; see Figure 2) exhibits decreased performance, primarily because the small dataset size did not sufficiently benefit the model during the continued pre-training phase, ultimately leading to a performance drop.

In contrast, GitTables (approximately 1000 data points and 9 features on average per dataset; see Figure 2) leads to performance improvements. The biggest performance improvements are achieved with our manually curated OpenML and Kaggle datasets (10k to 100k data points and on average tens of features). We intentionally chose a smaller, curated set of datasets from OpenML and Kaggle to effectively prevent data contamination, which is why we did not combine them with GitTables or CommonCrawl datasets.

6. Conclusion and Future Work

We show that continued pre-training of TabPFNv2 on curated, real-world tabular data yields a stronger default model, Real-TabPFN, which we will open-source. Bridging the synthetic-to-real gap, Real-TabPFN outperforms the default TabPFNv2 on most of the datasets and outperforms every other state-of-the-art baseline on all evaluated datasets. Additional experiments deliver the same message: seeing *more context*—whether temporal (longer windows) or statistical (a richer mix of bigger datasets) during continued pre-training produces larger improvements.

Acknowledgements

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 539134284, through EFRE (FEIH_2698644) and the state of Baden-Württemberg. We also acknowledge funding by the European Union (via ERC Consolidator Grant DeepLearning 2.0, grant no. 101045765). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. L.P. acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under SFB 1597 (SmallData), grant number 499552394; and funding by ELSA – European Lighthouse on Secure and Safe AI funded by the European Union under Grant Agreement No. 101070617. F.H. acknowledges the financial support of the Hector Foundation. Finally, we thank the reviewers for their feedback, which has helped us improve the manuscript.

Baden-Württemberg

Co-funded by the European Union

References

Bizer, C., Meusel, R., Lehmberg, O., Ritze, D., and Zope, S. Web data commons - web table corpus 2015 / english-language relational subset, 2015. URL https://madata.bib.uni-mannheim.de/208/.

Breiman, L. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. URL http://arxiv.org/abs/1603.02754.

Cox, D. R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B*, 20(2): 215–242, 1958.

den Breejen, F., Bae, S., Cha, S., and Yun, S.-Y. Fine-tuned in-context learning transformers are excellent tabular data classifiers, 2025. URL https://arxiv.org/abs/2405.13396.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dorogush, A. V., Gulin, A., Gusev, G., Kazeev, N., Prokhorenkova, L. O., and Vorobev, A. Fighting biases with dynamic boosting. *CoRR*, abs/1706.09516, 2017. URL http://arxiv.org/abs/1706.09516.

D'souza, A., Swetha, M., and Sarawagi, S. Synthetic tabular data generation for imbalanced classification: The surprising effectiveness of an overlap class. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16127–16134, 2025.

Eggert, G., Huo, K., Biven, M., and Waugh, J. Tablib: A dataset of 627m tables with context, 2023. URL https://arxiv.org/abs/2310.07875.

Gardner, J., Perdomo, J. C., and Schmidt, L. Large scale transfer learning for tabular data via language modeling, 2024. URL https://arxiv.org/abs/2406.12031.

Gijsbers, P., Bueno, M. L. P., Coors, S., LeDell, E., Poirier, S., Thomas, J., Bischl, B., and Vanschoren, J. Amlb: an automl benchmark, 2023. URL https://arxiv.org/abs/2207.12560.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on tabular data?, 2022. URL https://arxiv.org/abs/2207.08815.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. Tabpfn: A transformer that solves small tabular classification problems in a second, 2023. URL https://arxiv.org/abs/2207.01848.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Hulsebos, M., Demiralp, Ç., and Groth, P. Gittables: A large-scale corpus of relational tables. *CoRR*, abs/2106.07258, 2021. URL https://arxiv.org/abs/2106.07258.

Jiang, M., Liu, K. Z., Zhong, M., Schaeffer, R., Ouyang, S., Han, J., and Koyejo, S. Investigating data contamination for pre-training language models, 2024. URL https://arxiv.org/abs/2401.06059.

Kelly, M., Longjohn, R., and Nottingham, K. The uci machine learning repository, 2007. URL https://archive.ics.uci.edu.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114 (13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Li, X., Grandvalet, Y., and Davoine, F. Explicit inductive bias for transfer learning with convolutional networks, 2018. URL https://arxiv.org/abs/1802.01483.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization, 2017a.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017b. URL https://openreview.net/forum?id=Skq89Scxx.

Ma, J., Thomas, V., Hosseinzadeh, R., Kamkari, H., Labach, A., Cresswell, J. C., Golestan, K., Yu, G., Volkovs, M., and Caterini, A. L. Tabdpt: Scaling tabular foundation models, 2024. URL https://arxiv.org/abs/2410.18164.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Penedo, G., Kydlíček, H., allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. V., and Wolf, T. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Qu, J., Holzmüller, D., Varoquaux, G., and Morvan, M. L. Tabicl: A tabular foundation model for in-context learning on large data, 2025. URL https://arxiv.org/abs/2502.05564.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL http://arxiv.org/abs/1910.10683.

Tran, Q. M., Hoang, S. N., Nguyen, L. M., Phan, D., and Lam, H. T. Tabularfm: An open framework for tabular foundational models, 2024. URL https://arxiv.org/abs/2406.09837.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.264119.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yin, P., Neubig, G., tau Yih, W., and Riedel, S. Tabert: Pretraining for joint understanding of textual and tabular data, 2020. URL https://arxiv.org/abs/2005.08314.

A. Training Datasets

The following table lists the 71 datasets curated for continued pre-training, along with their source and access link.

Name	Source
aam_avaliacao_dataset	Kaggle
Air Traffic Data	Kaggle
ansible-defects-prediction	Kaggle
AV Healthcare Analytics II	Kaggle
Candidate Selection	Kaggle
Cardio Disease	Kaggle
CC_Fraud_Dataset	Kaggle
Classification - Crop Damages in India (2015-2019)	Kaggle
CSGO Round Winner Classification	Kaggle
Flower Type Prediction Machine Hack	Kaggle
Horse Racing - Tipster Bets	Kaggle
How severe the accident could be	Kaggle
HR Analysis Case Study	Kaggle
HR analysis	Kaggle
hr-comma-sep	Kaggle
ip-network-traffic-flows-labeled-with-87-apps	Kaggle
Janatahack cross-sell prediction	Kaggle
JanataHack Machine Learning for Banking	Kaggle
L&T Vehicle Loan Default Prediction	Kaggle
League of Legends Diamond Games (First 15 Minutes)	Kaggle
Multiple target variable classification - Hackathon	Kaggle
Online News Popularity	Kaggle
Online Shopper's Intention	Kaggle
Phishing website Detector	Kaggle
Phishing websites Data	Kaggle
Pump it Up Data Mining the Water Table	Kaggle
Richter's Predictor Modeling Earthquake Damage	Kaggle
Server Logs - Suspicious	Kaggle
Sloan Digital Sky Survey DR14	Kaggle
Sloan Digital Sky Survey DR16	Kaggle
Success of Bank Telemarketing Data	Kaggle
Term Deposit Prediction Data Set	Kaggle
trajectory-based-ship-classification	Kaggle
Travel Insurance	Kaggle
Amazon_employee_access	OpenML
artificial-characters	OpenML
Bank_marketing_data_set_UCI	OpenML
BNG(breast-w)	OpenML
BNG(tic-tac-toe)	OpenML
Click_prediction_small	OpenML
CreditCardSubset	OpenML
connect_4	OpenML
eeg-eye-state	OpenML
electricity	OpenML
elevators	OpenML
Employee-Turnover-at-TECHCO	OpenML
eye_movements	OpenML
FOREX_eurpln-hour-High	OpenML
gas-drift-different-concentrations	OpenML

Name	Source
gas-drift	OpenML
higgs	OpenML
house_16H	OpenML
house_8L	OpenML
Intersectional-Bias-Assessment-(Training-Data)	OpenML
law-school-admission-binary	OpenML
magic	OpenML
MagicTelescope	OpenML
Medical-Appointment	OpenML
microaggregation2	OpenML
fried	OpenML
mozilla4	OpenML
mushroom	OpenML
NewspaperChurn	OpenML
nursery	OpenML
okcupid_stem	OpenML
pendigits	OpenML
PhishingWebsites	OpenML
pol	OpenML
WBCAtt	OpenML
Bank Marketing	OpenML
Internet Firewall Data	OpenML

B. Evaluation Datasets

We use the same evaluation suite as TabPFNv2 to ensure direct comparability of results. All classification tasks from the AutoML Benchmark with fewer 10,000 samples and 500 features. The benchmark comprises diverse real-world tabular datasets, curated for complexity, relevance, and domain diversity.

Name	OpenML ID	Domain	Features	Samples	Targets	Categorical Feats.
ada	41156	Census	48	4147	2	0
Australian	40981	Finance	14	690	2	8
blood-transfusion-service-center	1464	Healthcare	4	748	2	0
car	40975	Automotive	6	1728	4	6
churn	40701	Telecommunication	20	5000	2	4
cmc	23	Public Health	9	1473	3	7
credit-g	31	Finance	20	1000	2	13
dna	40670	Biology	180	3186	3	180
eucalyptus	188	Agriculture	19	736	5	5
first-order-theorem-proving	1475	Computational Logic	51	6118	6	0
GesturePhase Segmentation Processed	4538	Human-Computer Interaction	32	9873	5	0
jasmine	41143	Natural Language Processing	144	2984	2	136
kc1	1067	Software Engineering	21	2109	2	0

Name	OpenML ID	Domain	Features	Samples	Targets	Categorical Feats.
kr-vs-kp	3	Game Strategy	36	3196	2	36
madeline	41144	Artificial	259	3140	2	0
mfeat-factors	12	Handwriting Recognition	216	2000	10	0
ozone-level-8hr	1487	Environmental	72	2534	2	0
pc4	1049	Software Engineering	37	1458	2	0
philippine	41145	Bioinformatics	308	5832	2	0
phoneme	1489	Audio	5	5404	2	0
qsar-biodeg	1494	Environmental	41	1055	2	0
Satellite	40900	Environmental Science	36	5100	2	0
segment	40984	Computer Vision	16	2310	7	0
steel-plates-fault	40982	Industrial	27	1941	7	0
sylvine	41146	Environmental Science	20	5124	2	0
vehicle	54	Image Classification	18	846	4	0
wilt	40983	Environmental	5	4839	2	0
wine-quality-white	40498	Food and Beverage	11	4898	7	0
yeast	181	Biology	8	1484	10	0

C. Performance Comparison on 29 AMLB Classification Datasets

Scores are normalized on all the baselines (0 = worst, 1 = best) per dataset; all methods are tuned for ROC-AUC, so secondary metrics may not reflect their true rank.

	Mean Normalized					Mean					Mean
	ROC ($\uparrow$)	Acc. ($\uparrow$)	F1 ($\uparrow$)	CE ($\downarrow$)	ECE ($\downarrow$)	ROC ($\uparrow$)	Acc. ($\uparrow$)	F1 ($\uparrow$)	CE ($\downarrow$)	ECE ($\downarrow$)	Time (s)
Real-TabPFN	**0.976** ±**0.01**	**0.932** ±**0.01**	**0.939** ±**0.01**	**0.011** ±**0.00**	**0.107** ±**0.01**	**0.932** ±**0.01**	**0.862** ±**0.02**	**0.771** ±**0.04**	0.337 ±0.03	**0.040** ±**0.01**	2.921 ±0.57
TabPFN (default)	0.954 ±0.01	0.906 ±0.01	0.920 ±0.01	0.036 ±0.01	0.111 ±0.02	0.929 ±0.01	0.857 ±0.02	0.767 ±0.04	0.347 ±0.03	0.042 ±0.01	2.793 ±0.49
Autogluon(V1, BQ) (tuned)	0.928 ±0.01	0.888 ±0.02	0.916 ±0.01	0.040 ±0.01	0.108 ±0.01	0.926 ±0.02	0.856 ±0.02	0.769 ±0.04	**0.311** ±**0.03**	0.041 ±0.01	9660.060 ±514.65
XGB (tuned)	0.842 ±0.02	0.748 ±0.02	0.759 ±0.02	0.268 ±0.03	0.367 ±0.03	0.920 ±0.02	0.844 ±0.02	0.739 ±0.04	0.432 ±0.08	0.066 ±0.03	14444.307 ±11.99
CatBoost (tuned)	0.832 ±0.02	0.776 ±0.02	0.790 ±0.02	0.186 ±0.02	0.285 ±0.03	0.920 ±0.02	0.844 ±0.02	0.741 ±0.04	0.408 ±0.06	0.057 ±0.02	14437.103 ±4.79
LightGBM (tuned)	0.781 ±0.02	0.720 ±0.03	0.767 ±0.02	0.252 ±0.03	0.361 ±0.04	0.915 ±0.02	0.841 ±0.02	0.741 ±0.04	0.443 ±0.11	0.063 ±0.02	14410.417 ±1.37
CatBoost (default)	0.761 ±0.02	0.731 ±0.02	0.783 ±0.02	0.170 ±0.02	0.249 ±0.02	0.913 ±0.02	0.839 ±0.02	0.748 ±0.04	0.404 ±0.04	0.053 ±0.01	5.874 ±0.74
Random Forest (tuned)	0.727 ±0.02	0.650 ±0.03	0.644 ±0.03	0.376 ±0.04	0.462 ±0.03	0.913 ±0.02	0.834 ±0.02	0.716 ±0.05	0.386 ±0.07	0.074 ±0.02	14404.904 ±0.15
LightGBM (default)	0.693 ±0.03	0.684 ±0.03	0.747 ±0.03	0.307 ±0.03	0.407 ±0.04	0.908 ±0.02	0.836 ±0.02	0.745 ±0.04	0.461 ±0.06	0.068 ±0.02	0.583 ±0.06
XGB (default)	0.665 ±0.03	0.643 ±0.03	0.725 ±0.03	0.330 ±0.03	0.533 ±0.04	0.906 ±0.02	0.834 ±0.02	0.743 ±0.04	0.468 ±0.06	0.079 ±0.02	0.814 ±0.09
Random Forest (default)	0.640 ±0.04	0.633 ±0.03	0.672 ±0.03	0.553 ±0.04	0.425 ±0.03	0.907 ±0.02	0.833 ±0.02	0.727 ±0.04	0.432 ±0.19	0.073 ±0.02	0.488 ±0.03
SVM (tuned)	0.571 ±0.03	0.531 ±0.03	0.537 ±0.03	0.292 ±0.03	0.169 ±0.02	0.887 ±0.02	0.810 ±0.02	0.680 ±0.04	0.455 ±0.04	0.044 ±0.01	14412.047 ±3.05
MLP (default)	0.512 ±0.03	0.442 ±0.03	0.480 ±0.03	0.345 ±0.03	0.294 ±0.03	0.883 ±0.02	0.802 ±0.02	0.664 ±0.05	0.493 ±0.05	0.058 ±0.02	2.133 ±0.19
MLP (sklearn) (tuned)	0.458 ±0.03	0.411 ±0.03	0.448 ±0.03	0.432 ±0.04	0.306 ±0.03	0.877 ±0.02	0.800 ±0.02	0.653 ±0.06	0.764 ±0.65	0.059 ±0.02	14408.730 ±0.34
Log. Regr. (tuned)	0.401 ±0.04	0.354 ±0.03	0.391 ±0.04	0.386 ±0.04	0.241 ±0.03	0.874 ±0.02	0.789 ±0.02	0.637 ±0.04	*inf* ±0.03	0.049 ±0.02	14406.416 ±0.47
SVM (default)	0.388 ±0.04	0.406 ±0.03	0.430 ±0.03	0.357 ±0.04	0.202 ±0.02	0.872 ±0.02	0.794 ±0.02	0.672 ±0.04	0.482 ±0.03	0.046 ±0.01	2.887 ±0.60
Log. Regr. (default)	0.209 ±0.03	0.185 ±0.03	0.186 ±0.03	0.483 ±0.04	0.348 ±0.04	0.857 ±0.02	0.778 ±0.02	0.600 ±0.04	0.529 ±0.03	0.062 ±0.02	0.609 ±0.10